

to judge pro

THE HONORABLE JOHN C. COUGHENOUR

*[Signature]*  
FILED \_\_\_\_\_ ENTERED \_\_\_\_\_  
LODGED \_\_\_\_\_ RECEIVED \_\_\_\_\_

APR 28 2003 PM

AT SEATTLE  
CLERK U.S. DISTRICT COURT  
WESTERN DISTRICT OF WASHINGTON  
BY \_\_\_\_\_ DEPUTY



CV 03 00077 #00000036

UNITED STATES DISTRICT COURT  
WESTERN DISTRICT OF WASHINGTON  
AT SEATTLE

MAILBLOCKS, INC., a California corporation,

Plaintiff,

v.

SPAM ARREST LLC, a Washington limited  
liability company,

Defendant

NO CV03-0077C

**DECLARATION OF JOHN LUTHER  
(COUNSEL FOR DEFENDANT) RE  
SUPPLEMENTAL AUTHORITY IN  
SUPPORT OF DEFENDANT SPAM  
ARREST LLC'S MOTION FOR  
PRELIMINARY INJUNCTION**

JOHN P. LUTHER declares as follows

1 I am over the age of eighteen, counsel for Defendant Spam Arrest LLC ("Spam Arrest") in this action and competent to testify as to the matters set forth in this declaration

**BACKGROUND**

2. As one of the attorneys at Newman & Newman who represents Spam Arrest, I became aware of Mailblocks, Inc. ("Mailblocks")'s Motion for Preliminary Injunction on March 7, 2003.

3 In preparing to oppose Mailblocks's Motion for Preliminary Injunction Spam Arrest requested a continuance so as to allow Spam Arrest to perform additional discovery. The Court denied Spam Arrest's request in a March 28 Order, but noted that it would consider that

**DECLARATION OF JOHN LUTHER  
(COUNSEL FOR DEFENDANT) re SUPPLEMENTAL  
AUTHORITY IN SUPPORT OF DEFENDANT'S  
MOTION FOR CONTINUANCE OF PL.'S MOT. FOR  
PRELIM. INJ.- 1  
Case No CV03-0077C**

NEWMAN & NEWMAN, ATTORNEYS AT LAW, LLP  
1001 Fourth Avenue Plaza, Suite 2560  
Seattle, Washington 98154  
phone (206) 624-6334  
fax (206) 624-6348

**ORIGINAL**

*36*

1 Spam Arrest has only a limited opportunity to conduct discovery in ruling on the Motion for  
 2 Preliminary Injunction

3 4 The law firm of Perkins Coie LLP had commissioned a patent search in advising  
 4 Spam Arrest with respect to its patent portfolio (prior to my contact with the matter) The search,  
 5 however, appeared to be incomplete and did not include certain references to prior art

6 5 After learning about the case, I immediately set out to perform research regarding  
 7 the validity of United States (*i e* , prior art references) 6,199,102 ("Cobb") and Patents 6,112,227  
 8 ("Heiner")

9 6 In the course of such research I performed Internet searches regarding the prior art  
 10 in the field I also commissioned a formal search through the firm of Express Search, Inc , a  
 11 Virginia-based company that performs patent searches ("Express Search") Those searches have  
 12 been ongoing to this date

13 7 A standard search performed by Express Search takes three (3) weeks On  
 14 Monday April 11, 2003 I contacted Express Search via e-mail to perform an expedited search for  
 15 information regarding the validity of the Cobb and Heiner Patents

16 8 On April 22, 2003, Express Search returned via fax the results of the expedited  
 17 search performed by them Attached as **Exhibit A** is the cover letter sent by Express Search to  
 18 me regarding the search results

19 9 During the pendency of the Express Search search, I performed additional  
 20 searches regarding prior art We did not have adequate time or resource time to perform these  
 21 searches prior to filing Spam Arrest's initial response to the Motion to Preliminary Injunction  
 22 Given the limited time provided to Spam Arrest for discovery, we still have not been able to do a  
 23 thorough search with respect to prior art Given some additional time we would be able to find  
 24 additional information germane to the validity of the Cobb and Heiner Patents

## 25 SUPPLEMENTAL PRIOR ART

26 10 Attached as **Exhibit B** is a true and correct copy of a article published and made  
 27 publicly available in 1992 titled "Pricing via Processing or Combatting Junk Mail," authored by  
 28

1 Cynthia Dwork and Moni Naor According to the article, a preliminary version of the article was  
2 presented at "Crypto 1992," a 1992 conference

3 11 Attached as **Exhibit C** is a true and correct copy of an article published and made  
4 available in 1996 titled "Verification of a human in the Loop or Identification via the Turing  
5 Test," authored by Moni Naor The article outlines the "Turing Test."

6 12 Attached as **Exhibit D** is a true and correct copy of an article dated August 17,  
7 1997, titled "Does Parallel Repetition Lower the Error in Computationally Sound Protocols,"  
8 authored by Mihir Bellare, Russell Impagliazzo and Moni Naor

9 13 Attached as **Exhibit E** is a true and correct copy of a Microsoft Research SVC  
10 Power Point presentation titled "Fighting Spam May be Easier Than you Think," authored by  
11 Cynthia Dwork of Microsoft Research SVC and found at the following URL  
12 <http://www.cis.upenn.edu/~spyce/presentations/Cynthia-Sep-02.pdf>

13 14 Attached as **Exhibit F** is a true and correct copy of an article by Janet Kornblum,  
14 dated August 6, 1997, titled "Programmer Writes Spam Bomb," and published at news.com (on-  
15 line at <http://news.com.com/2100-1023-202165.html>)

16 15 Attached as **Exhibit G** is a true and correct copy of an article by News.com staff,  
17 dated February 8, 1997, titled "ISP Internet Spam Provider," and published at news.com (on-line  
18 at <http://news.com.com/2100-1023-271857.html>)

19 16 Attached as **Exhibit H** is a true and correct copy of a posting to a newsgroup by  
20 Guy Tyler discussing the Deadbolt Personal E-mail Filter The newsgroup posting appears to  
21 have been made on July 16, 1997, and given additional time to conduct discovery, Spam Arrest  
22 will verify that this is the case

23 17 Attached as **Exhibit I** is a true and correct copy of an article by Andrew Leonard,  
24 dated September 21, 1997, titled "Spam Bombers," and published at salon.com (on-line at  
25 <http://archive.salon.com/sept97/21st/spam970904.html>).

26 18. Attached as **Exhibit J** is a true and correct copy of a posting to a newsgroup by  
27 Fred Elbel discussing the Waffle E-mail Spam Filter The newsgroup posting appears to have  
28

1 been made on April 25, 1997, and given additional time to conduct discovery, Spam Arrest  
2 could verify that this is the case

3 19 Attached as **Exhibit K** is a true and correct copy of a posting to a newsgroup by  
4 Ian Stirling discussing the idea that an effective spam filter would require a question to be  
5 answered that can only be answered using human intelligence The newsgroup posting appears  
6 to have been made on February 20, 1997, and given additional time to conduct discovery, Spam  
7 Arrest could verify that this is the case

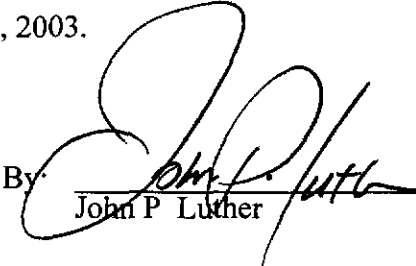
8 20 Attached as **Exhibit L** is a true and correct copy of a posting to a newsgroup by  
9 Brandon Hume discussing an effective spam filter which would verify that the e-mail in question  
10 is being sent by a human being and not a computer. The newsgroup posting appears to have been  
11 made on November 15, 1997, and given additional time to conduct discovery, Spam Arrest  
12 could verify that this is the case

13 21 Attached as **Exhibit M** is a true and correct copy of a printout from  
14 Mailcircuit com's website (at <[www.mailcircuit.com/filter.htm](http://www.mailcircuit.com/filter.htm)>) describing Mailcircuit's  
15 handshake verification system

16 22. Attached as **Exhibit N** is a true and correct copy of two posting to newsgroups by  
17 Julian Byrne, discussing details of the "challenge response" method of blocking spam The  
18 newsgroup posting appears to have been made on January 28, 1997, and given additional time to  
19 conduct discovery, Spam Arrest could verify that this is the case

20  
21 I declare under penalty of perjury of the laws of the State of Washington that the  
22 foregoing is true and correct to the best of my knowledge

23 DATED this 28<sup>th</sup> day of April, 2003.

24  
25  
26 By  \_\_\_\_\_  
27 John P. Luther  
28

**DECLARATION OF JOHN LUTHER**  
**(COUNSEL FOR DEFENDANT) re SUPPLEMENTAL**  
**AUTHORITY IN SUPPORT OF DEFENDANT'S**  
**MOTION FOR CONTINUANCE OF PL.'S MOT. FOR**  
**PRELIM. INJ.- 4**  
Case No CV03-0077C

**NEWMAN & NEWMAN, ATTORNEYS AT LAW, LLP**  
1001 Fourth Avenue Plaza, Suite 2560  
Seattle, Washington 98154  
phone (206) 624-6334  
fax (206) 624-6348





## EXPRESS SEARCH, INC.

2121 Eisenhower Ave., Suite 500 Alexandria, Virginia 22314  
Phone: (703) 535-5455 • Fax: (703) 535-5475  
Patent and Trademark Services

Facsimile Transmission

Date: 4/22/03

To: Newman + Newman

Attention: Mr. Luther / Mr. Newman

Fax Number: ( ) :

Phone Number: ( ) :

From: Penelope

Re: Rush Validity + Literature Searches

Comments:

Your results are being overnighted via UPS.  
Your tracking # is 1Z F2R 326 014001 3590.  
Thank you!

Number of Pages (including cover sheet): 12

If you do not receive this complete transmission, please call the office of Express Search, Inc. at once, at (703) 535-5455

This facsimile contains confidential information, intended only for the use of the recipient named above. If you are not the intended recipient, you are hereby notified that any dissemination or copying of this communication is strictly prohibited. If you have received this communication in error, please call (703) 535-5455 at once.

**EXPRESS SEARCH, INC.**

2121 Eisenhower Ave., Suite 500 Alexandria, VA 22314  
Phone: (703) 535-5455 - Fax: (703) 535-5475

*Patent and Trademark Services*  
*www.ExpressSearch.com*

Newman & Newman  
1001 Fourth Avenue Plaza  
Suite 2560  
Seattle, WA 98154

Re: Patent Validity Search  
Control Of E-mail Spam

April 22, 2003

Dear John,

In accordance with your fax received April 16, 2003, a Patent Validity Search of 28 hours was conducted at the U.S. Patent and Trademark Office for U.S. Patents 6,112,227 and 6,199,102, in accordance with the disclosure provided.

The following Examiners were consulted regarding the field of search:

Examiner Dinh in Art Unit 2153

The following classes and subclasses were searched:

Class 709 (Electrical Computers And Digital Processing Systems: Multiple Computer Or Process Coordinating)

Subs. 103, 200, 206, 207, 203, 202, 204, 205, 216, 225, 245, 314, 238

Class 713 (Electrical Computers And Digital Processing Systems: Support)

Subs. 155, 200

Class H04L (Transmission Of Digital Information)

Subs. 09/00, 09/32, 12/00

Class H04M (Telephonic Communication)

Subs. 01/64, 03/42, 01/64

Class 379 (Telephonic Communication)

Subs. 88.26

Class 395 [ABOLISHED BY PTO AND REFERENCES TRANSFERRED TO CLASS 709]

Subs. 200.32, 200.363, 200.36, 200.51, 200.03

The following U.S. patents were noted as being most relevant:

|           |           |           |           |           |           |           |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 6,052,709 | 6,023,723 | 5,999,967 | 5,930,479 | 5,909,589 | 5,884,033 | 5,859,967 |
| 5,835,722 | 5,826,022 | 5,796,840 | 5,781,857 | 5,742,769 | 5,734,903 | 5,655,079 |
| 5,632,018 | 5,630,123 | 5,627,764 | 5,619,648 | 5,608,786 | 5,604,803 | 5,600,799 |
| 5,548,789 | 5,471,519 | 5,448,734 | 5,423,042 | 5,377,354 | 5,333,266 | 5,283,856 |
| 5,245,532 | 5,159,673 | 5,093,918 | 5,040,141 | 4,977,520 |           |           |

The following foreign patents were also noted of interest:

|            |            |            |            |
|------------|------------|------------|------------|
| CN 1117680 | IL 0120023 | EP 0760565 | BP 0725523 |
| EP 0721268 | EP 0686327 | EP 0651533 | EP 0463252 |
| WO 9837675 | WO 9726709 | WO 9724825 | WO 9723082 |
| WO 9720423 | WO 9714234 | WO 9624213 | WO 9609714 |
| WO 9406236 |            |            |            |

The following patents appear to be most relevant:

WO 9609714 discloses a personal communications internetworking (40), which provides a network subscriber with the ability to remotely control the receipt and delivery of wireless and wireline messages; and said network operates as an interface, and it also performs media translation, just as the receipt and delivery options are maintained in a database including screening, rejection lists, and selective destination delivery of incoming e-mail (Figure 1; Claim 1; Abstract).

EP 0686327 discloses a method of handling a message or document, which includes means for directing said message or document (2) to a trusted sealing device; and said device is characterized in that said sealing device displays said message or document to a human user for visual checking, and the user may decide that the message is incorrect or he/she no longer wants to send the message and, accordingly, actuates a predetermined switch on the trusted sealing device to reject the message (Abstract; Claim 1).

5,471,519 discloses a communications monitoring and control system, which includes methods and apparatus for processing a communication directed to a called party; and upon receipt of a communication, the called party may accept, reject or monitor the communication (Abstract; Claim 1; Figure 1).

WO 9726709 discloses a multi-function electronic messaging system (18), which includes means for transmitting messages over a first communications link (24) having a remote unit (22) and e-mail screening criteria; and first computer sends an acknowledgement signal back to the user of the remote unit (22) over the first communications link (24) and decodes and error checks the signals (Figure 1; Claim 1; Abstract).

WO 9720423 discloses a system and method for managing incoming calls, which allow a



subscriber to have incoming telephone calls automatically screened and directed to a subscriber with a minimum of interruptions and which includes various systems for registration of those attempting to communicate (Figure 1; Claim 1; Abstract).

4,977,520 discloses an electronic calendaring method for screening e-mail messages, which includes an interactive selection means belonging to the calendar owner end user of at least one option which can be either to accept the invitation and calendar the meeting or to reject the invitation (Figure 1; Claim 1; Abstract).

The following patents also appear to be of interest:

6,052,709 discloses a system and method and system for controlling delivery of unsolicited electronic mail messages, which includes mailboxes corresponding to the spam probe of e-mail addresses which addresses are monitored for incoming mail by a spam control center (Abstract; Claim 1; Figure 1).

6,023,723 discloses a system and method of filtering junk e-mails, which includes an address as well as specific character strings included in e-mail transmitted to a central location to be included in a master list and the master list is periodically sent to each of the users allowing the first filter to be updated (Figure 1; Claim 1; Abstract).

5,999,967 discloses a method for filtering email, which includes a sender side part with responsibility for attaching the electronic stamp; and a receiver side part with responsibility for removing the electronic stamp and filtering the electronic mail based on the value of the electronic stamp (Figure 1; Claim 1; Abstract).

5,930,479 discloses a system and method for sending and receiving authorized messages from a sender to a recipient in a network, which comprises a common address portion that indicates the identity of the recipient in the network and a channel identifier portion for verifying that the message is authorized for delivery to the recipient (Figure 1; Claim 1; Abstract).

5,909,589 discloses a verification process, which includes data captured by the habit capture system and which is provided to a verifier for sampling the user's characteristics and comparing the characteristics of the current user with that stored in a database (Figure 1; Claim 1; Abstract).

5,884,033 discloses a client-based filtering system, which compares portions of incoming and/or outgoing messages to filtering information in a filter database (Figure 1; Claim 1; Abstract).

5,859,967 discloses a method and system for authorized users, which includes means of registration and the user transmits, from a registered account, an email message which includes the fax number to which the facsimile is to be transmitted (Figure 1; Claim 1; Abstract).

5,835,722 discloses a computer terminal and a method for blocking the use and transmission of vulgar and pornographic material, which comprises a responsive and interactive manner that

comprehensively monitors computer operations for creation or transmission of vulgar and pornographic material (Figure 1; Claim 1; Abstract).

5,826,022 discloses an apparatus for handling of electronic mail messages, which provides users with a mechanism for ignoring a particular ongoing e-mail discussion until that ongoing discussion terminates (Figure 1; Claim 1; Abstract).

5,796,840 discloses a hardware agent, which comprises a device-specific key pair within the hardware agent and verifies the key pair is unique (Figure 1; Claim 1; Abstract).

5,781,857 discloses a method of establishing an e-mail monitor external to the wireless communication system, which comprises accessing the wireless communication system; and selecting an email service option (Figure 1; Claim 1; Abstract).

5,742,769 discloses a directory service, which allows a user to receive email messages from senders without requiring the user to reveal his/her email address (Figure 1; Claim 1; Abstract).

5,734,903 discloses a system for object oriented message filtering, which includes means for selectively transferring a message between a client task and one or more server tasks for preprocessing, processing, and postprocessing comprises an object database having a filter object memory, an object management unit, a message transaction unit, and a locking unit (Figure 1; Claim 1; Abstract).

5,655,079 discloses a sending computer protocol, which transmits data with either address or content code assigned, and the computers other than the sending computer decide whether or not to receive the data according to either the address or the content code (Figure 1; Claim 1; Abstract).

5,632,018 discloses an electronic mail system, which comprises broadcasting means for sending a message with broadcast addresses received from a first user to users with said broadcast addresses; and response receiving means for receiving a response to said message from a second user who received said message from said broadcasting means and the like (Figure 1; Claim 1; Abstract).

5,630,123 discloses a software system (2) utilizing a filtered priority queue (10), which includes a filtering module (4) operable to access a plurality of data records of entities (16, 18, 20, 22, 24, 26 and 28) of a priority queue and to filter and arrange the data records in a memory storage device (8) to form the filtered priority queue (10).

5,627,764 discloses a rule based electronic messaging system, which includes a controller utility that accesses a user-list-file with the user name of every "user" whose messages are to be automatically, periodically processed; and upon accessing the user-list-file, a user name is selected and that user's copy of the mail messaging facility is started (Figure 1; Claim 1; Abstract).

5,619,648 discloses a technique for reducing the amount of junk e-mail received by a user

of an e-mail system, which uses non-address information and model information to determine whether the e-mail message should be provided to the user (Figure 1; Claim 1; Abstract).

5,608,786 discloses a unified messaging system, which includes means for receiving mail, E-mail, facsimiles and other message types for retrieval by the subscriber; and data communication networks such as the Internet can become global voice mail and facsimile mail systems (Figure 1; Claim 1; Abstract).

5,604,803 discloses a method for user authentication between a first computer and a second computer, which includes a client workstation that provides a login address as an anonymous ftp (file transfer protocol) request, and a password as a user's e-mail address; and the destination server then sends the PEM encrypted password random number, as an ftp file, over the Internet to the client workstation (Figure 1; Claim 1; Abstract).

5,600,799 discloses an interface system, which includes means for transferring information between a local area-network and a system memory associated with a station attached to the network and which includes a bus interface unit for implementing the transfer of information between the interface system and the memory system and it includes a request module (Abstract; Claim 1; Figure 1).

5,548,789 discloses a message communication processing system, which includes a plurality of communication terminals for processing media information and a center apparatus having a storage and forward exchange function for communication messages sent from the communication terminals (Figure 1; Claim 1; Abstract).

5,448,734 discloses a method and apparatus for communicating messages, which includes means for sending and message receiving processes; and it receives messages of certain classes or types generated by the system or by other applications and the remote application may register itself with the message server (102) by means of a suitable system call (Figure 1; Claim 1; Abstract).

5,423,042 discloses a registration type-mail server system, which executes the client code without re-compiling or re-linking and clients provide the server with executable object or registration code along with information about the code that allows the server to register the code object (Figure 1; Claim 1; Abstract).

5,377,354 discloses a method and apparatus for prioritizing messages, which includes a comparator (52) which matches keywords and which rules are subject to revision by members of the group establishing rules (Abstract; Claim 1; Figure 1).

5,333,266 discloses an Integrated Messaging System which integrates mail from a plurality of mail servers handling messages of different media types such as text, voice, facsimile, video and image; and it maintains one in-basket for all mail systems, eliminating the need to collect each type of mail separately and a reject class (Figure 1; Claim 1; Abstract).

5,283,856 discloses a flexible, event driven and conditional rule based mail messaging system, which includes a rule mechanism, a conditional, action-invoking paradigm or "triplet" which permits definition of a repertoire of events considered to be significant events upon which to trigger actions in the electronic mail messaging system (Figure 1; Claim 1; Abstract).

5,245,532 discloses an e-mail data processing system, which includes a mail follow-up system for monitoring and processing selected mail items, a facility for tracking the mail, facility for storing said characteristic data into tag fields, an administrator system for centrally administering the tagged mail on a departmental basis (Figure 1; Claim 1; Abstract).

5,159,673 discloses a control system for an e-mail system, which includes a system task memory and a global mailbox register associated with said system task memory, as well as means for receiving unsolicited ones of said messages from said programmable logic controller directed to said global mailbox via said communications network (Claims 3, 4; Abstract; Figure 1).

5,093,918 discloses an attribute list, which may contain multiple sets of attributes and multiple sets of users and each end user may periodically determine and display the status of an individual mail object with regard to the entire group of recipients, a subgroup of recipients or an individual end user (Figure 1; Claim 1; Abstract).

5,040,141 discloses a method for administering reply mail in electronic mail system, which includes a table for indicating mail messages requesting answers and means for registering mail which requires an answer on said table (Figure 1; Claim 1; Abstract).

CN 1117680 discloses an information security intelligence system, which includes an intelligence communication platform, a message transmission platform, fourth generation electronic mail, database and such software as multiple access control, data cipher/decipher, electronic digital signature and verification (Figure 1; Claim 1; Abstract).

IL 0120023 discloses a notification method for e-mail, which includes notifying recipient of at least one new e-mail message addressed to a subscriber, and includes providing to the subscriber, through the telephone network, notification on the existence of at least one new e-mail message (Abstract; Claim 1; Figure 1).

EP 0760565 discloses an apparatus for authenticating information, which has been sent by a sender via a dispatcher to a recipient and which includes means for securing at least part of said authentication-information against undetected tamper attempts of at least said sender (Figure 1; Claim 1; Abstract).

EP 0725523 discloses a method for routing messages, which includes defining the origin and destination addresses with a stack of nested multi-element address specifications, and editing, at least one intermediate node in said networks, said stacked and nested multi-element address specifications and the user agent 132 will send a return message back to the originating point terminal. (Figure 1;

Claim 1; Abstract).

EP 0721268 discloses a method of routing a message in an electronic mail system, which comprises a plurality of user selectable action items, means for defining a reply address and a forwarding address; and providing to the user the option of selecting between reply addresses and forwarding addresses (Claim 1; Abstract).

EP 0651533 discloses a method for providing secure communications, which includes first data processing device and second device and comprises transmitting a first message which includes a Mobile Certificate including a mobile public key, a chosen challenge value (CH1), and a list of supported shared key algorithms (SKCS) (Abstract; Claim 1; Figure 1).

EP 0463252 discloses a data processing apparatus, which includes means for executing a computer program having its own message handling routine for handling messages generated by the computer program and comprising a plurality of objects (2, 4, 6, 8, 10, 12) including data and code for manipulating said data, the apparatus having a system messaging means for transferring messages between objects (Figure 1; Claim 1; Abstract).

WO 9837675 discloses a method for the secure transmission of data certificate, which includes means for authenticating the identity of a party sending an e-mail; and a name-value pair is transmitted to an administrative function on the third computer (140) and is routed to the appropriate certification authority; and it transmits other certification information from said administrative function to said certification authority on the second computer (Abstract; Claim 1; Figure 1).

WO 9724825 discloses a method for the automatic, secure and direct transmission of data, which comprises means for the transmission of e-mail and calls for the data to be transmitted from a first terminal (1) to a first microcomputer system (11) which is directly associated with the first terminal (1).

WO 9723082 discloses an e-mail message system, which includes means of connecting a telephone network (30) and Internet network (20) via a guest-mail server (40); and it receives a telephone call from a data terminal (31) in the telephone network and via the call receives a fax message containing a destination address in the Internet network (Figure 1; Claim 1; Abstract).

WO 9714234 discloses point-to-point Internet protocol exchanges, which includes means for exchanging Internet Protocol (IP) addresses between units to establish a point-to-point communication links; and in response to identification of one of the entries by a requesting user processor, providing the network protocol address of the identified entry to the requesting user process (Figure 1; Claim 1; Abstract).

WO 9624213 discloses an interconnected free e-mail system, which may include commands and options that are selectable by the user, and can further include logo, artwork and/or information about a particular subscriber communication network (Abstract; Claim 1; Figure 1).

APR-22-2003 TUE 03:22 PM EXPRESS SEARCH INC

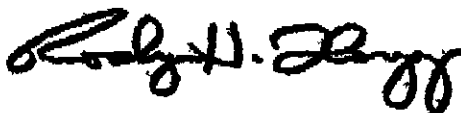
7035355475

P. 09

WO 9406236 discloses a personal number communications system, which assigns a personal number to each subscriber, and receives from each subscriber communication routing information, and said info includes one or more hierarchical lists of destinations based on the time of day and day of the week (Figure 1; Claim 1; Abstract).

These patents are representative of the prior art searched. Copies of the cited prior art are enclosed for your further review. Please do not hesitate to contact me with any questions regarding this search.


Best Regards,  
EXPRESS SEARCH

A handwritten signature in black ink, appearing to read "Rodger H. Flagg".

Rodger Flagg  
President

RHF/dtc  
Enclosure: 50 Patents  
Ref: N110-V1523



|  <b>EXPRESS SEARCH INC.</b><br><b>GRAPHING REPORT</b>                                                                                                                                                                                                                                                                                     |          |          |          |          |  |  |  |  |  |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------|----------|----------|--|--|--|--|--|--|
| <b>Patent No. 6,199,102</b><br><b>Claim No. 1</b><br><b>Claim: A method for filtering electronic messages, the method comprising:</b>                                                                                                                                                                                                                                                                                      |          |          |          |          |  |  |  |  |  |  |
| <b>Project No. 31523</b>                                                                                                                                                                                                                                                                                                                                                                                                   |          |          |          |          |  |  |  |  |  |  |
| <b>FEATURES</b>                                                                                                                                                                                                                                                                                                                                                                                                            |          |          |          |          |  |  |  |  |  |  |
| <b>A</b> Receiving an electronic message from a sender, the message including an address field containing a sender's address;<br><b>B</b> Comparing the sender's address to a list of accepted senders;<br><b>C</b> Sending a prompt back to the sender if the sender's address is not contained in the list of accepted senders,<br><b>D</b> Wherein the prompt is designed to be answered by a person and not a machine. |          |          |          |          |  |  |  |  |  |  |
| <b>PATENT NUMBER</b>                                                                                                                                                                                                                                                                                                                                                                                                       | <b>A</b> | <b>B</b> | <b>C</b> | <b>D</b> |  |  |  |  |  |  |
| EP0686327                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        |          | x        |  |  |  |  |  |  |
| 5,377,354                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        |          |          |  |  |  |  |  |  |
| 5,909,589                                                                                                                                                                                                                                                                                                                                                                                                                  |          | x        |          |          |  |  |  |  |  |  |
| 5,930,479                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        |          |          |  |  |  |  |  |  |
| 5,999,967                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        |          | x        |  |  |  |  |  |  |
| 6,023,723                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        |          | x        |  |  |  |  |  |  |
| WO9837675                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        | x        |          |  |  |  |  |  |  |
| 5604803                                                                                                                                                                                                                                                                                                                                                                                                                    | x        | x        | x        |          |  |  |  |  |  |  |
| EP0725523                                                                                                                                                                                                                                                                                                                                                                                                                  |          |          | x        |          |  |  |  |  |  |  |
| IL0120023                                                                                                                                                                                                                                                                                                                                                                                                                  | x        |          |          |          |  |  |  |  |  |  |
| WO9714234                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        | x        |          |  |  |  |  |  |  |
| 5471519                                                                                                                                                                                                                                                                                                                                                                                                                    | x        | x        |          |          |  |  |  |  |  |  |
| WO9406236                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        |          |          |  |  |  |  |  |  |
| EP0651533                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        | x        | x        |  |  |  |  |  |  |
| WO9726709                                                                                                                                                                                                                                                                                                                                                                                                                  | x        | x        |          | x        |  |  |  |  |  |  |
| 4977520                                                                                                                                                                                                                                                                                                                                                                                                                    | x        | x        |          | x        |  |  |  |  |  |  |
| 5283856                                                                                                                                                                                                                                                                                                                                                                                                                    | x        | x        |          | x        |  |  |  |  |  |  |
| 5627764                                                                                                                                                                                                                                                                                                                                                                                                                    |          |          | x        |          |  |  |  |  |  |  |
|                                                                                                                                                                                                                                                                                                                                                                                                                            |          |          |          |          |  |  |  |  |  |  |
|                                                                                                                                                                                                                                                                                                                                                                                                                            |          |          |          |          |  |  |  |  |  |  |


\*This graphing report is intended as a guide to aid patent counsel in reviewing the relevance of the cited prior art and is not intended to be a legal opinion of the relevance of the prior art uncovered in the search.

APR-22-2003 TUE 03:23 PM

EXPRESS SEARCH INC

7035355475

P. 11

|  <b>EXPRESS SEARCH INC.</b><br><b>GRAPHING REPORT</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |   |   |   |   |   |  |  |  |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---|---|---|---|---|--|--|--|--|
| <b>Patent No. 6,112,227</b><br><b>Claim No. 12</b><br><b>Project No. 31523</b><br><b>Claim: A method for preventing the delivery of unwanted electronic mail messages to a destination client comprising the steps of:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |   |   |   |   |   |   |  |  |  |  |
| <b>FEATURES</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |   |   |   |   |   |   |  |  |  |  |
| <b>A</b> Receiving an original electronic mail message addressed to a destination client from a source client having an electronic mail address at a destination server;<br><b>B</b> Comparing said electronic mail address of said source client to an accept list of electronic mail addresses;<br><b>C</b> Sending said original electronic mail message to said destination client when said electronic mail address of said source client is on said accept list;<br><b>D</b> Comparing said electronic mail address of said source client to a reject list of electronic mail addresses when said electronic address of said source client is not on said accept list;<br><b>E</b> Deleting said original message when said electronic mail address of said source client is on said reject list;<br><b>F</b> Sending a reply electronic mail message from said destination server to said source client requesting that said source client complete a registration process when said electronic mail address of said source client is not on said reject list; |   |   |   |   |   |   |  |  |  |  |
| PATENT NUMBER                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | A | B | C | D | E | F |  |  |  |  |
| EP0686327                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x |   |   |   |   |   |  |  |  |  |
| 5,377,354                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |   | x |   |   |   |   |  |  |  |  |
| 5,859,967                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x |   | x |   |   | x |  |  |  |  |
| 5,909,589                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x | x | x |   |   |  |  |  |  |
| 5,930,479                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x |   | x |   |   |  |  |  |  |
| 5,999,967                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x |   | x |   |   |  |  |  |  |
| 6023723                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | x | x | x | x | x |   |  |  |  |  |
| EP1059779                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x | x | x | x | x |  |  |  |  |
| WO9837675                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x | x | x | x |   |  |  |  |  |
| 5604803                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | x | x | x | x | x |   |  |  |  |  |
| EP0725523                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x |   | x |   |   |  |  |  |  |
| WO9714234                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x | x |   |   |   |  |  |  |  |
| 5471519                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | x | x | x |   |   |   |  |  |  |  |
| WO9406236                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x | x | x |   |   |  |  |  |  |
| EP0651533                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x | x | x |   |   |  |  |  |  |
| WO9726709                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x | x | x | x | x |   |  |  |  |  |
| 5283856                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | x |   | x |   | x |   |  |  |  |  |
| WO9720423                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | x |   |   |   |   | x |  |  |  |  |
| 5627764                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | x |   | x |   | x | x |  |  |  |  |

\*This graphing report is intended as a guide to aid patent counsel in reviewing the relevance of the cited prior art and is not intended to be a legal opinion of the relevance of the prior art uncovered in the search



EXPRESS SEARCH INC

7035355475

\*This graphing report is intended as a guide to aid patent counsel in reviewing the relevance of the cited prior art and is not intended to be a legal opinion of the relevance of the prior art uncovered in the search.



# Pricing via Processing or Combatting Junk Mail\*

Cynthia Dwork \*      Moni Naor †

Draft of full version

## Abstract

We present a computational technique for combatting junk mail, in particular, and controlling access to a shared resource, in general. The main idea is to require a user to compute a moderately hard, but not intractable, function in order to gain access to the resource, thus preventing frivolous use. To this end we suggest several *pricing functions*, based on, respectively, extracting square roots modulo a prime, the Fiat-Shamir signature scheme, and the Ong-Schnorr-Shamir (cracked) signature scheme.

---

\*A Preliminary version of this paper was presented at Crypto'92

\*IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, CA 95120. E-mail: dwork@almaden.ibm.com

†Incumbent of the Morris and Rose Goldman Career Development Chair, Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel. Most of this work performed while at the IBM Almaden Research Center. Research supported by an Alon Fellowship and a grant from the Israel Science Foundation administered by the Israeli Academy of Sciences. E-mail: naor@wisdom.weizmann.ac.il

# 1 Introduction

Some time ago one of us returned from a brief vacation, only to find 241 messages in our reader. While junk mail has long been a nuisance in hard (snail) mail, we believe that electronic junk mail presents a much greater problem. In particular, the ease and low cost of sending electronic mail, and in particular the simplicity of sending the same message to many parties, all but invite abuse. In this paper we suggest a computational approach to combatting the proliferation of electronic mail. More generally, we have designed an *access control mechanism* that can be used whenever it is desirable to restrain, but not prohibit, access to a resource.

Two general approaches have been used for limiting access to a resource: legislation and usage fees. For example, it has been suggested that sending an unsolicited FAX message should be a misdemeanor. This approach encounters obvious definitional problems. Usage fees may be a deterrent, however, we do not want a system in which to send a letter or note between friends should have a cost similar to that of a postage stamp, similarly we do not wish to charge a high fee to transmit long files between professional collaborators. Such an approach could lead to *underutilization* of the electronic medium.

Since we believe the real cost of using the medium (plus the profit to the provider) will not serve as a deterrent to junk mail, we propose a system that imposes another type of cost on transmissions. These costs will deter junk mail but will not interfere with other uses of the system. The main idea is for the mail system to require the sender to compute some moderately expensive, but not intractable, function of the message and some additional information. Such a function is called a *pricing function*.

In the more general setting, in which we have an arbitrary resource and a resource manager, a user desiring access to the resource would compute a moderately hard function of the *request id*. (The request id could be composed of the user's identifier together with, say, the date and time of the request.)

The pricing function may be chosen to have something like a trap door: given some additional information the computation would be considerably less expensive. We call this a *shortcut*. The shortcut may be used by the resource manager to allocate cheap access to the resource as the manager sees fit, by bypassing the control mechanism. For example, in the case of electronic mail the shortcut permits the post office to grant bulk mailings at a price chosen by the post office, circumventing the cost of directly evaluating the pricing function for each recipient.

We believe our approach to be of practical interest. It also raises the point that, unlike the situation with one-way functions (functions that are easy to compute but hard to invert) and Cryptography, there is virtually no complexity theory of moderately hard functions, and therefore yields excellent motivation for the development of such a theory.

The rest of this paper is organized as follows. Section 2 contains a description of the properties we require of pricing functions. Section 3 focusses on combatting junk

mail. Section 4 describes three possible candidates for pricing functions. We require a family of hash functions satisfying certain properties. Potentially suitable hash functions are discussed in more detail in Section 5. Section 6 contains conclusions and open problems.

## 2 Definitions and Properties

We must distinguish between several grades of difficulty of computation. Rather than describe the hardness of computing a function in terms of asymptotic growth, or in terms of times on a particular machine, we focus on the *relative* difficulty of certain computational tasks.

We require three classes of difficulty: *easy*, *moderate*, and *hard*. The term *moderate* can be viewed in two different ways. As an upper bound, it means that computation should be at *most* moderately hard (as opposed to hard), as a lower bound it means that computation should be at *least* moderately easy (as opposed to easy). The precise definition of easy and moderate and hard will depend on the particular implementation. However, there must be some significant gap between easy and moderately easy. As usual, *hard* means intractable in reasonable time, such as factoring a 1024-bit product of two large primes.

The functions we consider for implementing our scheme have a *difference parameter* that serves a role analogous to that of a *security parameter* in a cryptosystem. A larger difference parameter stretches the difference between easy and moderate. Thus, if it is desired that, on a given machine, checking that a function has been correctly evaluated should require only, say,  $10^{-2}$  seconds of CPU time, while evaluating the function directly, without access to the shortcut information, should require 10 seconds, the difference parameter can be chosen appropriately.

A function  $f$  is a *pricing function* if

- 1  $f$  is moderately easy to compute,
- 2  $f$  is not amenable to amortization: given  $\ell$  values  $m_1, \dots, m_\ell$ , the *amortized* cost of computing  $f(m_1), \dots, f(m_\ell)$  is comparable to computing  $f(m_i)$  for any  $1 \leq i \leq \ell$ ,
- 3 given  $x$  and  $y$  it is easy to determine if  $y = f(x)$ .

We use the term “function” loosely: sometimes  $f$  will be a relation. That is, given  $x$  it should be moderately easy to find a  $y$  such that the pair  $(x, y)$  satisfies the relation, but given  $(x, y)$  it should be easy to determine whether it satisfies the relation.

Let  $S \subseteq \{0, 1\}^*$  be a set that can be easily sampled (i.e. there is an efficient algorithm for selecting a random  $s \in S$ ).  $F = \{f_s | s \in S\}$  is a *family* of pricing functions indexed by  $s \in S \subseteq \{0, 1\}^*$  if, given  $s$ ,  $f_s$  is a pricing function. We will be interested in a collection of families of pricing function  $\mathcal{F} = \{F_k | k \geq 1\}$ , indexed by

a difference parameter  $k$ , where the hardness of evaluating  $f_s \in F_k$  should increase with  $k$

**Remark 2.1** *It is important not to choose a function that after some preprocessing can be computed very efficiently. Consider the following family of pricing functions  $F$ , based on subset sum. The index  $s$  is a set of  $\ell$  numbers  $a_1, a_2, \dots, a_\ell$ ,  $1 \leq a_i \leq 2^\ell$ , such that  $2^\ell$  is moderately large. For a given request  $x$ ,  $f_s(x)$  is a subset of  $a_1, a_2, \dots, a_\ell$  that sums to  $x$ . Computing  $f_s$  seems to require time proportional to  $2^\ell$ . As was shown by Schroepel and Shamir [24], after preprocessing, using only a moderate amount of storage, such problems can be solved much more efficiently. Thus, there could be large difference between the time spent evaluating  $f_s$  on a large number  $k$  of different inputs, such as would be necessary for sending bulk mail, and  $k$  individual computations of  $f_s$  from scratch. This is clearly undesirable.*

We now introduce the notion of a *shortcut*, similar in spirit to a trapdoor one-way permutation, introduced by Diffie and Hellman [10]. A pricing function with a shortcut is easy to evaluate given the shortcut. In particular, the shortcut is used for bypassing the access control mechanism, at the discretion of the resource manager.

A collection  $\mathcal{F}$  of families of pricing functions is said to have the *shortcut property* if for  $k \geq 1$  there exists an efficient algorithm that generates a pair  $(s, c)$  where

- 1  $s$  is uniformly distributed in  $S$
- 2 given  $s$  (but not  $c$ )  $f_s$  is a function in  $\mathcal{F}$
- 3  $c$  is a *shortcut* computing  $f_s$  is easy given  $s$  and  $c$

Note that since  $f_s$  is a pricing function, it is not amenable to amortization. Thus, given  $s$ , finding  $c$  or an equivalent shortcut, should be hard.

**Remark 2.2** *The consequences of a “broken” function are not severe. For example, if a cheating sender actually sends few messages, then little harm is done, if it sends many messages then the cheating will be suspected, if not actually detected, and the pricing function or its key can be changed.*

In the context of junk mail we use hash functions so that we never apply the pricing function to a message, which may be long, but only to its hash value. Ideally, the hash function should be very easy to compute. However, given  $m$ ,  $h$ , and  $m'$ , it should not be easy to find  $m''$  closely related to  $m'$  such that  $h(m'') = h(m)$ . For example, if Macy’s sends an announcement  $m$  of a sale, and later wishes to send an announcement  $m'$  of another sale, it should not be easy to find a suffix  $z$  such that  $h(m' \parallel z) = h(m)$ .

Suitable hash functions could be based on DES, subset sum, MD4, MD5, and Snefru. We briefly discuss each of these in Section 5.

### 3 Junk Mail

The primary motivation for our work is combatting electronic junk mail. We envision an environment in which people have computers that are connected to a communication network. The computers may be used for various anticipated activities, such as, for example, updating one's personal database (learning that a check has cleared), subscribing to a news service, and so on. This communication requires no human participation. This is different from the situation when one receives a personal letter, or an advertisement of a product in which one is likely to be interested, which clearly require one's attention. Our interest is in controlling mail of this second kind.

The system requires a single pricing function  $f_s$ , with shortcut  $c$ , and a hash function  $h$ . The selection of the pricing function and the setting of usage fees are controlled by a *pricing authority*. All users agree to obey the authority. There can be any number of trusted agents that receive the shortcut information from the pricing authority. The functions  $h$  and  $f_s$  are known to all users, but only the pricing authority and its trusted agents know  $c$ .

To send a message  $m$  at time  $t$  to destination  $d$ , the sender computes  $y = f_s(h(\langle m, t, d \rangle))$  and sends  $\langle y, m, t \rangle$  to  $d$ . The recipient's mail program verifies that  $y = f_s(h(\langle m, t, d \rangle))$ . If the verification fails, or if  $t$  is significantly different from the current time, then the message is discarded and (optionally) the sender is notified that transmission failed. If the verification succeeds and the message is timely, then the message is routed to the reader.

Suppose the pricing function  $f$  has no short-cut. In this case, if one wants to write a personal letter, the computation of  $f_s$  may take time proportional to the time taken to compose the letter. For typical private use that may be acceptable. In contrast, the computational cost of a bulk mailing, even a "desirable" (not junk) mailing, would be prohibitive, defeating the whole point of high bandwidth communication.

In our approach bulk mail such as a call for papers for a professional conference, or an announcement of a new product, is sent using the shortcut  $c$ , which necessarily requires the participation of the system manager. The sender pays a fee and prepares a set of letters, and one of the trusted agents evaluates the pricing function as needed for all the letters, using the shortcut. Since the fee is levied to deter junk mail, and not to cover the actual costs of the mailing, it can simply be turned over to the recipients of the message (and used to pay for the services of the authority).

Finally, each user can have a *frequent correspondent list* of senders from whom messages are accepted without verification. Thus, friends and relatives could circumvent the system entirely. Moreover, one could join a mailing list by adding the name of the distributor to one's list of frequent correspondents<sup>1</sup>. The list, which is maintained locally by the recipient, can be changed as needed. Thus, when submitting a paper to a conference, an author can add the name of the conference to the list of frequent corresponders. In this way the conference is spared the fees of bulk mailing.

<sup>1</sup>Similarly, one could have a list of senders to whom access is categorically denied.



## 4 Pricing Functions

In this section we list three candidate families of pricing functions. All the candidates use number-theoretic algorithms. For a good introduction to this area see [9, 13]. The first pricing function is the simplest, but has no shortcut. The other two do have good shortcuts.

### 4.1 Extracting Square Roots

The simplest implementation of our idea is to base the difficulty of sending on the difficulty (but not infeasibility) of extracting square roots modulo a prime  $p$ . Again, there is no known shortcut for this function.

- **Index:** A prime  $p$  of length depending on the difference parameter, a reasonable length would be 1024 bits.
- **Definition of  $f_p$ :** The domain of  $f_p$  is  $Z_p$ .  $f_p(x) = \sqrt{x} \bmod p$ .
- **Verification:** Given  $x, y$ , check that  $y^2 \equiv x \bmod p$ .

The checking step requires only one multiplication. In contrast, no method of extracting square roots mod  $p$  is known that requires fewer than about  $\log p$  multiplications. Thus, the larger we take the length of  $p$ , the larger the difference between the time needed to evaluate  $f_p$  and the time needed for verification.

### 4.2 A Fiat-Shamir Based Scheme

This pricing function described in this section is based on the signature scheme of Fiat and Shamir [11]. The idea is to reduce the difficulty of forging signatures in that scheme. The security of the Fiat-Shamir signature Scheme is based on

- The difficulty of factoring large numbers (or equivalently of extracting square-roots modulo a composite)
- A hash function whose range size is (exponential in) the security parameter. Ideally, this hash function should behave as a random function and the time it takes to forge a message should be proportional to the range size.

The proposed pricing function is obtained by taking the Fiat-Shamir signature scheme with a smaller security parameter for the hash function. Searching a range of size exponential in the security parameter should be feasible, but time-consuming. The scheme is as follows.



- **Index:** Let  $N = pq$ , where  $p$  and  $q$  are primes of sufficient length to make factoring  $N$  infeasible (currently 512 bits each suffices, but if there is further progress in factoring algorithms, then 1024 bits should be used). Let  $y_1 = x_1^2, \dots, y_k = x_k^2$  be  $k$  squares modulo  $N$ , where  $k$  depends on the difference parameter. Finally, let  $h$  be a hash function whose domain is  $Z_N^* \times Z_N^*$ , and whose range is  $\{0, 1\}^k$ .  $h$  can be obtained from any of the hash functions described in Section 5 by taking the  $k$  least significant bits of the output. The index  $s$  is the  $(k+2)$ -tuple  $(N, y_1, \dots, y_k, h)$ .
- **Shortcut:** The square roots  $x_1, \dots, x_k$ .
- **Definition of  $f_s$ :** The domain of  $f_s$  is  $Z_N^*$ . Below, we describe a moderately easy algorithm for finding  $z$  and  $r^2$  satisfying the following conditions. Let us write  $h(x, r^2) = b_1 \dots b_k$ , where each  $b_i$  is a single bit. Then  $z$  and  $r^2$  must satisfy

$$z^2 = r^2 x^2 \prod_{i=1}^k y_i^{b_i} \pmod{N}.$$

$$f_s(x) = (z, r^2) \text{ (note that } f_s \text{ is a relation)}$$

- **Verification:** Given  $x, z, r^2$ , compute  $b_1 \dots b_k = h(x, r^2)$  and check that

$$z^2 = r^2 x^2 \prod_{i=1}^k y_i^{b_i} \pmod{N}$$

- **To Evaluate  $f_s$  with Shortcut Information:** Choose an  $r$  at random, compute  $h(x, r^2) = b_1 \dots b_k$ , and set  $z = rx \prod_{i=1}^k x_i^{b_i}$ .  $f_s(x) = (z, r^2)$ .
- **Evaluating  $f_s$  without Shortcut Information:**

$f_s(x) = (z, r^2)$  can be computed as follows

Guess  $b_1 \dots b_k \in \{0, 1\}^k$   
 Compute  $B = \prod_{i=1}^k y_i^{b_i} \pmod{N}$   
 Repeat  
     Choose random  $z \in Z_N^*$   
     Define  $r^2$  to be  $r^2 = (z^2/Bx^2) \pmod{N}$   
 Until  $h(x, r^2) = b_1 \dots b_k$

In the evaluation of  $f_s$  without the shortcut the expected number of iterations is  $2^k$ , which, based on the intuition driving the Fiat-Shamir signature scheme, seems to be the best one can hope for. In particular, if  $h$  is random, then one can do no better. In particular, retrieving the shortcut  $x_1, \dots, x_k$  is as hard as factoring [21]. In contrast, the verification procedure involves about  $k$  multiplications (actually  $k/2+1$  expected multiplications) and one evaluation of the hash function. Similarly, given

the shortcut the function can be evaluated using about  $k$  multiplications and one evaluation of the hash function. Thus,  $k$  is the difference parameter. A reasonable choice is  $k = 10$ .

### 4.3 An Ong-Schnorr-Shamir Based Scheme, or, Recycling Broken Signature Schemes

A source of suggestions for pricing functions with short cuts is signature schemes that have been broken. The “right” type of breaking applicable for our purposes is one that does not retrieve the private signature key (analogous to factoring  $N$  in the previous subsection), but nevertheless allows forging signatures by some moderately easy algorithm.

In this section we describe an implementation based on the proposed signature scheme of Ong, Schnorr and Shamir and the Pollard algorithm for breaking it. In [18, 19] Ong, Schnorr, and Shamir suggested a very efficient signature scheme based on quadratic equations modulo a composite: the public key is a modulus  $N$  (whose factorization remains secret) and an element  $\ell \in Z_N^*$ . The private key is  $u$  such that  $u^2 = -\ell^{-1} \bmod N$ , (i.e. a square root of the inverse of  $-\ell$  modulo  $N$ ). A signature for a message  $m$  (which we assume is in the range  $0 \leq m < N$ ) is a solution  $(x_1, x_2)$  of the equation  $x_1^2 + \ell x_2^2 = m \bmod N$ . There is an efficient signing algorithm, requiring knowledge of the private key

- choose random  $r_1, r_2 \in Z_N^*$  such that  $r_1 - r_2 = m \bmod N$
- set  $x_1 = \frac{1}{2} (r_1 + r_2) \bmod N$  and  $x_2 = \frac{1}{2} u (r_1 - r_2) \bmod N$

Note that verifying a signature is extremely easy, requiring only 3 modular multiplication.

Pollard (reported and extended in [20]) suggested a method of solving the equation without prior knowledge of the private key (finding the private key itself is hard – equivalent to factoring [21]). The method requires roughly  $\log N$  iterations, and thus can be considered moderately hard, as compared with the verification and signing algorithms, which require only a constant number of multiplications and inversions. For excellent descriptions of Pollard’s method and related work see [6, 14].

We now describe how to use the Ong-Schnorr-Shamir signature scheme as a pricing function.

- **Index:** Let  $N = pq$  where  $p$  and  $q$  are primes let  $\ell \in Z_N^*$ . Then  $s = (N, \ell)$
- **Shortcut:**  $u$  such that  $u^2 = \ell^{-1} \bmod N$
- **Definition of  $f_s$ :** The domain of  $f_s$  is  $Z_N^*$ . Then  $f_s(x) = (x_1, x_2)$ , where  $x_1^2 + \ell x_2^2 = x \bmod N$ .  $f_s$  is computed using Pollard’s algorithm, as described above.

- **Verification:** Given  $x_1, x_2, x$  verify that  $x = x_1^2 + \ell x_2^2$
- **To Evaluate  $f_s$  with Shortcut Information:** Use the Ong-Schnorr-Shamir algorithm for signing

## 5 Hash Functions

Recall that we need hash functions for two purposes. First, in the context of junk mail, we hash messages down to some reasonable length, say 512 bits, and apply the pricing function to the hashed value of the message. In addition, we need hashing in the pricing function based on the signature scheme of Fiat-Shamir.

We briefly discuss four candidate hash functions. Each of these can be computed very quickly.

- **DES:** Several methods have been suggested for creating a one-way hash function based on DES (e.g. [16] and the references contained therein). Since DES is implemented in VLSI, and such a chip might become widely used for other purposes, this approach would be very efficient. Note that various attacks based on the “birthday paradox” [8] are not really relevant to our application since the effort needed to carry out such attacks is moderately hard.
- **MD4 & MD5 :** MD4 and MD5 are candidate one-way hash functions proposed by Rivest [22, 23]. They were designed explicitly to have a high speed *software* implementation and are in wide use. The length of the output is either 128 or 256 bits. Although a simplified version of MD4 has been successfully attacked [3], we know of no attack on the full MD4. Also, [4] finds “pseudo-collisions” in MD5, but it is not clear whether this can be converted into a collision finding algorithm.
- **Subset Sum:** Impagliazzo and Naor [12] have proposed using “high density” subset sum problems as one-way hash functions. They showed that finding colliding pairs is as hard as solving the subset sum problem for this density. Although this approach is probably less efficient than the others mentioned here, the function enjoys many useful statistical properties (viz. [12]). Moreover, it is parameterized and therefore flexible.
- **Snefru:** Snefru was proposed by Merkle [17] as a one-way hash function suitable for software, and was broken by Biham and Shamir [2]. However, the Biham and Shamir attack still requires about  $2^{24}$  operations to find a partner of a given message. Thus, it may still be viable for our purposes.

## 6 Discussion and Further Research

Of the three pricing functions described in Section 4, the Fiat-Shamir is the most flexible and enjoys the greatest difference function — changing  $k$  by 1 doubles the difference. The disadvantage is that this function, like the Fiat-Shamir scheme, requires the “extra” hash function.

As mentioned in the Introduction, there is no theory of moderately hard functions. The most obvious theoretical open question is to develop such a theory, analogous, perhaps, to the theory of one-way functions. Another area of research is to find additional candidates for pricing functions. Fortunately, a trial and error approach here is not so risky as in cryptography, since as discussed earlier, the consequences of a “broken” pricing function are not severe. If someone tries to make money from having found cheaper ways of evaluating the pricing function, then he or she underprices the pricing authority. Either few people will know about this, in which case the damage is slight, or it will become public.

A growing area of research is the economics of networks [15, 7, 5] where issues such as the effect of pricing on the network behavior are investigated. It is interesting to see whether there are connection between this direction and the ideas suggested in this paper.

Finally, the evaluation of the pricing function serves no useful purpose, except serving as a deterrent. It would be exciting to come up with a scheme in which evaluating the pricing function serves some additional purpose.

## References

- [1] T. A. Berson, *Differential cryptanalysis mod  $2^{32}$  with applications to MD5*, Advances in Cryptology — Eurocrypt '92, Springer-Verlag, 1992.
- [2] E. Biham and A. Shamir, *Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI, and Lucifer*, Advances in Cryptology - Proceedings of Crypto'91, Lecture Notes in Computer Science No. 576, Springer Verlag, 1992, pp. 156–171.
- [3] B. den Boer and A. Bosselaers, *An attack on the last two rounds of MD4*, Advances in Cryptology - Proceedings of Crypto'91, Lecture Notes in Computer Science No. 576, Springer Verlag, 1992, pp. 194–203.
- [4] B. den Boer and A. Bosselaers, *Collisions for the compression function of MD5*, Advances in Cryptology — Eurocrypt '93, Springer-Verlag, Lecture Notes in Computer Science 765, 1994, pp. 293–304.
- [5] R. Bohn, H-W. Braun, K. C. Claffy and S. Wolff, *Mitigating the coming Internet crunch: multiple service levels via Precedence*, Journal on High Speed Networks, 1994.

- [6] E F Brickell and A M Odlyzko *Cryptanalysis A Survey of Recent Results*, Proceedings of the IEEE, vol 76, pp 578-593, May 1988
- [7] R Cochi, D Estrin, S Shenkar and L Zhang, *Pricing in computer networks, motivation, formulation, and Example*, ACM-IEEE Trans on Computer Networks
- [8] D Coppersmith, *Another Birthday Attack*, Advances in Cryptology - Proc Crypto '85, Springer Verlag, LNCS, Vol 218, pp 369-378
- [9] T H Cormen, C E Leiserson and R L Rivest, **Introduction to Algorithms**, MIT Press/McGraw-Hill, 1990
- [10] W Diffie and M E Hellman, *New Directions in Cryptography*, IEEE Trans Inform Theory, IT-22, 1976, pp 644-654
- [11] A Fiat and A Shamir, *How to prove yourself*, Advances in Cryptology - Proc of Crypto 86, pp 641-654
- [12] R Impagliazzo and M Naor, *Cryptographic schemes provably secure as subset sum*, Proc of the 30th IEEE Symp on Foundation of Computer Science 1989
- [13] N Koblitz, **A Course in Number Theory and Cryptography**, Springer-Verlag, New York, 1987
- [14] K McCurley, *Odd and ends from cryptology and computational number theory*, in **Cryptology and computational number theory**, edited by C Pomerance, AMS short course, 1990. pp 145-166
- [15] J Mackie-Mason and H Varian, *pricing the Internet*, Technical Report, U Michigan, 1993
- [16] R C Merkle *One Way Functions and DES*, Advances in Cryptology - Proc of Crypto'89, LNCS 435 Springer Verlag, 1990, pp 428-446
- [17] R C Merkle, *Fast Software One-Way Hash Function*, J of Cryptology Vol 3, No 1, pp 43-58, 1990
- [18] H Ong, C P Schnorr and A Shamir, *An efficient signature scheme based on quadratic equations*, Proc 16th ACM Symp of Theory of Computing, 1984, pp 208-216
- [19] H Ong, C P Schnorr and A Shamir, *Efficient signature scheme based on polynomial equations*, Advances in Cryptology - Proc of Crypto 84, LNCS 196. 1985, pp 37-46
- [20] J M Pollard and C P Schnorr, *Solution of  $X^2 + ky^2 = m \bmod n$* , IEEE Trans on Information Theory, 1988

- [21] M. O. Rabin, *Digital Signatures and Public Key Functions as Intractable as Factoring* Technical Memo TM-212, Lab. for Computer Science, MIT, 1979
- [22] R. L. Rivest, *The MD4 Message Digest Algorithm*, Advances in Cryptology - Proc of Crypto'90, LNCS 537, Springer Verlag, 1991, pp. 303–311
- [23] R. L. Rivest, *The MD5 Message Digest Algorithm*, Internet Request for Comments, April 1992, RFC 1321
- [24] R. Schroepel and A. Shamir, *A  $T = O(2^{n/2})$ ,  $S = O(2^{n/4})$  algorithm for certain NP-complete problems* SIAM J. Computing, 10 (1981), pp. 456–464



Verification of a human in the loop  
or  
Identification via the Turing Test\*

Moni Naor<sup>†</sup>

September 13th, 1996

**Abstract**

We propose using a “Turing Test” in order to verify that a human is the one making a query to a service over the web. Thus, before a request is processed the user should answer as a challenge an instance of a problem chosen so that it is easy for humans to solve but the best known programs fail on a non-negligible fraction of the instances. We discuss several scenarios where such tests are desired and several potential sources for problems instances. We also discuss the application of this idea for combatting junk mail.

---

\*A preliminary draft

<sup>†</sup>Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: naor@wisdom.weizmann.ac.il



## 1 Introduction

It is quite common now for various companies to offer services on the Web free of charge (or for a promotional fee). These include various search engines (Alta Vista, Infoseek, Inktomi, Lycos, Yahoo, etc ) and shopping arcades and catalogs. An emerging phenomena is that of “meta-services” a program that provides the user with an interface for requesting information (or perform comparative shopping). Given a request from the user, the program accesses several such services in parallel providing each of them with the request. It then processes the information obtained from these services and presents it to the user. Examples for such meta-services are MetaCrawler [7] and Savvy Search [9] for search engines and BargainFinder Agent [1] for comparative shopping. While such meta-services have many advantages for the user, from the service provider point of view they are not necessarily desirable. The reason such services are free is in order to build customer loyalty and for advertisements. Using their output for further processing defeats these purposes, since they lose the direct contact with the user. Therefore it may be desirable for such companies to have a method for forcing a human-being to fill in the forms for the request and not a program<sup>1</sup>. The goal of this paper is to present a scheme that may discourage (unauthorized) meta-services from filtering the user interface of the services they employ. There can be various other settings where one wants to let human users access the resource, but to exclude software robots, or give them a lower priority. One of them, combatting junk mail, is discussed below.

One of the key ideas of Cryptography is applying the fact that there are intractable problems, i.e. problems that cannot be solved effectively by any feasible machine, in order to construct secure protocols. Our proposal is to adapt the way identification is handled in cryptographic settings to deal with this situation. There, when one party  $\mathcal{A}$  wants to prove its identity to another party  $\mathcal{B}$ , the process is a proof that the  $\mathcal{A}$  can effectively compute a (keyed) function that a different user (not having the key) cannot compute. The identification process consists of a *challenge* selected by  $\mathcal{B}$  and the response computed by  $\mathcal{A}$ <sup>2</sup>.

What should replace the keyed cryptographic function in the current setting are those tasks where humans excel in performing, but machines have a hard-time competing with the performance of a three years old child. By performing such a task successfully the user’s proves that it is human. We envision the following scheme for applying this idea: when a service sends a form to be filled in with the user’s request it will also send a “human-in-the-loop-challenge” which will be one or several questions that can be answered easily by any person. When the user fills in his request he should also answer the questions provided as the challenge. Before the service processes the request it should verify the correctness of the answers. The service will not process a query whose attached questions were not answered properly (or will give it a lower priority). The questions for the challenge should be chosen from a large collection of possible questions and will be specific to the user’s request, i.e.

<sup>1</sup>An alternative is to ban certain addresses from using the service. The reason this is not sufficient is that in the near future the meta-service may run on the client’s machine.

<sup>2</sup>This description includes both the symmetric case where  $\mathcal{A}$  and  $\mathcal{B}$  share the common key that defines a function (the classical Identification-Friend-or-Foe) and the public-key setting where  $\mathcal{A}$ ’s function is defined by a public key so that anyone can verify the correctness, but no one but  $\mathcal{A}$  can compute successfully with high probability.

there should be no point in gathering those questions

We therefore want for our "Turing Tests" a collection of problems with the following properties

- 1 It is easy to generate many instances of the problem, together with their unambiguous solution. The method for generating the problems can be either internal, i.e. there is a generator that gets as input some random bits and outputs an instance of the problem, or based on external input, e.g. a video camera positioned in a crowded street. It is best if the method for generation does not require human intervention at all. However, if human assistance is needed for creating a model from which several instances are derived, then it is still reasonable
- 2 Humans can solve a given instance effortlessly with very few errors. Providing the answer should also be easy, e.g. typing a small number of characters.
- 3 The best known programs for solving such problems fail on a non-negligible fraction of the problems, even if the method of generating the instances is known. The number of instances in a challenge will depend on this fraction.
- 4 An instance specification is succinct both in the amount of communication needed to describe it and in the area it takes to present it to the user

## 2 Sources for the Turing Tests

We now list a few areas that are a possible source for such problems. They are drawn from Vision and natural language processing

- Gender recognition - given a picture of a face determine whether it is a male or a female. Since there are only two possibilities the challenge should consist of, say, four pictures and the users should get all of them right. Getting many different pictures for the collection from which the challenge is drawn does not seem difficult, but one should make sure that they are indeed easy for a human being
- Facial expression understanding - given a face decide whether it is happy or sad
- Find body parts - Benny Pinkas suggested that the challenge be a picture of, say, an animal and the user should click on its eye. The advantage over all other proposals here is that the number of possible answers is much larger. There should be of course some tolerance for the distance from the correct location
- Deciding nudity - given several pictures determine which one contains the undressed person. Here there is work done in [4] that reports much progress in this area
- Naive drawing understanding - given a drawing of, say, a house determine what it is from a list of five distinct possibilities. Dan Roth suggested adding "context", i.e. background, to the drawing - this will make it easier for people and harder for machines. Also break the lines

- Handwriting understanding - given a handwritten word the user should type it. Again, it makes sense to add the kind of noise that people do not have a problem to ignore.
- Speech recognition - the challenge is a recording of several words and the user should write them. Given progress in this area, selecting from several possibilities may be too easy, having the user write the result may be too demanding, since there are spelling errors etc.
- Filling in words - Given a sentence where the subject has been deleted and a list of words, select one for the subject. This can be generated more or less automatically from a large corpus. It has the advantage that it is more succinct to represent. However, yet again the progress in solving such problems automatically may be too advanced and using statistical methods is sufficient. Another possibility is to take a sentence and permute the order of the words. The challenge is to determine which of several possibilities is the original one.
- Disambiguation - another problem from NLP (suggested by Dan Roth). The challenge is to figure out to what does "it" refer in a sentence like "The dog killed the cat. It was taken to the morgue." The problem here is that it seems difficult to generate many different examples. Also it may be too demanding on the human user.

### 3 Remarks

Suppose that the meta-service develops a method that answers correctly 10% of the challenges. Then it would be tempted to try about 10 times in parallel, until it gets one of the challenges it can resolve. In order to prevent this, the set of instances should be chosen as a function of the query, i.e. will be given to the user only after he fills in the form. In any case, it is a good strategy for the service provider not to answer frequent queries from the same source and to follow whether the same query is given frequently in a given time slot.

The scheme suggested here works rather well with advertisements. The file containing the relevant puzzles may include also the advertisements, thus making filtering them difficult. Also the questions in the challenge may be somehow related to the content of the advertisement<sup>3</sup>.

Etzioni [3] calls search services like Alta Vista information *herbivores* and the meta-services information *carnivores*. Selberg and Etzioni [7] predict a state where the meta-services work together with the information gatherers by carrying out their advertisements or by sharing the profits. However, it is not clear what motivation the meta-services will have for this Isaiah like ideal [5], unless the herbivores will have the means of protecting themselves. The "Turing Test" approach proposed in this paper provides them with such means.

Another possible advantage of the scheme proposed is encouraging research in those areas that are chosen as challenges. One has to look at the problem of factoring numbers and see the tremendous algorithmic progress made there since it was suggested as a basis for cryptographic protocols [10] to realize the potential.

<sup>3</sup>Interestingly, a Berkeley company called Cybergold plans to offer "rewards" to people who follow links with advertisements.

Regarding some social issues concerning this proposal, it is possible to use in order to create culturally exclusive zones. This may have some advantages, e.g. for keeping children away, but in all likelihood is not very desirable. Therefore, care has to be taken when composing the tests so as not to make them culturally sensitive.

#### **4 Comparison with “Pricing via Processing”**

Dwork and Naor [2] proposed a method for combatting junk e-mail and in general for sharing resources when charging for them is either not economical or would not act as a proper deterrent. They suggested that in order for one user to send a message to another user the sender should compute a moderately hard function (taking, say, 20 seconds CPU time on a standard processor) of the content of the letter and the name of the addressee, thus way demonstrating that the receiver’s attention is important for the sender. The current proposal is also applicable for the junk mail scenario: to send a letter to a user, the sender sends the message and receives a challenge of the type described in the preceding sections that he should answer. The message is forwarded to the receiver’s attention only if the sender answers the challenge correctly. In this scenario it is important that the generation of the instances be free of human intervention.

The disadvantage of the Turing Test approach over the one described in [2] is that the protocol becomes a three round one (instead of a single round). Also the proposal of [2] provides for a “shortcut”, a way for computing the pricing function efficiently by a trusted agent, say for a reasonable price. This is useful for legitimate mass mailing, say invitations to a party (the deterrence in this case is the amount charged by the trusted agent).

The advantage over “pricing via processing” is that similar commodities are involved - to get the addressee’s attention the sender invests some of his own (human) time, and not his CPU time.

#### **5 Further Research**

The most intriguing direction of research this paper proposes is whether there are automated Turing Tests: can a computer be the interrogator, i.e. the player trying to establish whether the entity on the other end is a machine or a human. This should be the case even if the program being tested has access to the program of the interrogator (but not to private random bits used to generate problems).

#### **Acknowledgements**

we thank Cynthia Dwork, Benny Pinkas, Omer Reingold, Dan Roth and Shimon Ullman for useful remarks.

#### **References**

- [1] BargainFinder Agent (Andersen Consulting), available <http://bf.cstar.ac.com/bf>

- [2] C Dwork and M Naor, *Pricing via Processing or Combatting Junk Mail*, Advances in Cryptology – Proceedings of Crypto '92, Lecture Notes in Computer Science 740, Springer Verlag, 1993, pp 139–147
- [3] O Etzioni, *Moving up the information food chain: deploying Softbots on the World Wide Web*, Proc AAAI-96
- [4] M.M. Fleck, D A. Forsyth, C Bregler Finding Naked People, Proc 4th European Conf on Computer Vision, 1996
- [5] Isaiah, Chapter 11
- [6] Karov, Y , and S Edelman, Similarity-based word sense disambiguation, Weizmann Institute CS-TR 96-06, 1996
- [7] E Selberg and O Etzioni, *Multi-Service Search and Comparison using the MetaCrawler* Proc of the 4th International World Wide Web Conference Search available in [http //www metacrawler com](http://www.metacrawler.com)
- [8] Y Moses, D Reynard, and A Blake, 1995 Determining facial expressions in Real Time Proceeding of International Conference on Computer Vision p 296-301.
- [9] Savvy Search, search available in [http //guaraldi cs colostate edu.2000/form?beta](http://guaraldi.cs.colostate.edu.2000/form?beta)
- [10] A Odlyzko, The future of integer factorization, CryptoBytes (The technical newsletter of RSA Laboratories), 1.2, pp pp 5-12, 1995.
- [11] A M Turing, *Computing machinery and Intelligence*, Mind 59, 433–460, 1950.



An extended abstract of this paper appears in *Proceedings of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997. This is the full version.

## Does Parallel Repetition Lower the Error in Computationally Sound Protocols?

MIHIR BELLARE\*

RUSSELL IMPAGLIAZZO<sup>†</sup>

MONI NAOR<sup>‡</sup>

August 17, 1997

### Abstract

Whether or not parallel repetition lowers the error has been a fundamental question in the theory of protocols, with applications in many different areas. It is well known that parallel repetition reduces the error at an exponential rate in interactive proofs and Arthur-Merlin games. It seems to have been taken for granted that the same is true in arguments, or other proofs where the soundness only holds with respect to computationally bounded parties.

We show that this is not the case. Surprisingly, parallel repetition can actually fail in this setting. We present four-round protocols whose error does not decrease under parallel repetition. This holds for any (polynomial) number of repetitions. These protocols exploit non-malleable encryption and can be based on any trapdoor permutation. On the other hand we show that for three-round protocols the error does go down exponentially fast.

The question of parallel error reduction is particularly important when the protocol is used in cryptographic settings like identification, and the error represent the probability that an intruder succeeds.

---

\*Department of Computer Science & Engineering, Mail Code 0114, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-mail: [mihir@cs.ucsd.edu](mailto:mihir@cs.ucsd.edu). Supported in part by NSF CAREER Award CCR-9624439 and a 1996 Packard Foundation Fellowship in Science and Engineering.

<sup>†</sup>Department of Computer Science & Engineering, Mail Code 0114, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. E-mail: [russell@cs.ucsd.edu](mailto:russell@cs.ucsd.edu).

<sup>‡</sup>Incumbent of the Morris and Rose Goldman Career Development Chair, Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: [naor@wisdom.weizmann.ac.il](mailto:naor@wisdom.weizmann.ac.il). Supported in part by BSF Grant 32-00032-1.

## Contents

|          |                                                                       |           |
|----------|-----------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                   | <b>3</b>  |
| 1 1      | Our results                                                           | 4         |
| 1 2      | The bigger picture                                                    | 4         |
| 1 3      | A closer look                                                         | 5         |
| <b>2</b> | <b>Definitions and setup</b>                                          | <b>6</b>  |
| 2 1      | Computationally sound protocols                                       | 6         |
| 2 2      | Parallel repetition                                                   | 7         |
| 2 3      | Black-box amplification                                               | 7         |
| <b>3</b> | <b>Parallel repetition fails in general</b>                           | <b>8</b>  |
| 3 1      | Non-malleable encryption                                              | 8         |
| 3 2      | Two fold parallel repetition fails                                    | 8         |
| 3 3      | Many-fold parallel repetition fails                                   | 9         |
| 3 4      | Failure of parallel error reduction with low communication            | 10        |
| 3 5      | Extensions                                                            | 12        |
| <b>4</b> | <b>Parallel repetition reduces the error of three round protocols</b> | <b>12</b> |
| <b>5</b> | <b>Open problems and on-going work</b>                                | <b>13</b> |
| <b>6</b> | <b>Acknowledgments</b>                                                | <b>13</b> |
|          | <b>References</b>                                                     | <b>13</b> |
| <b>A</b> | <b>Proofs for Section 3</b>                                           | <b>15</b> |
| <b>B</b> | <b>Proof of Theorem 4.1</b>                                           | <b>18</b> |
| <b>C</b> | <b>Parallel repetition when the verifier has secret information</b>   | <b>23</b> |



# 1 Introduction

Various notions of interactive protocols [23, 3, 6, 8] have found wide applicability over the last decade. They have turned out to be essential for cryptography but somewhat more surprisingly, they have also been key to complexity theory, in particular to the theory of hardness of approximation problems (see [1] for a survey). For many of these applications, the purpose of the protocol is for a “verifier” to distinguish between a “good” prover making a legitimate claim and a “bad” prover attempting to trick the verifier into accepting incorrectly. An “error bound” or “error probability” for the protocol is a value for which we have a guarantee that any bad prover will be caught except perhaps with probability this value. Many natural constructions of protocols have relatively large error bounds (constant or worse), and most applications require error bounds that are small (less than  $1/\text{poly}$  for any polynomial  $\text{poly}$ ).

To bridge this gap, there are two generic methods of repeating protocols intended to reduce the error: sequential repetition and parallel repetition. Sequential repetition, repeating the protocol several times, beginning the next run after the previous one terminates, reduces error in all important models. It also preserves desirable properties of the original protocol, such as zero-knowledge (see [21, 30]). However, this is an expensive solution, in that it increases the number of communication rounds of the protocol, which is undesirable for both practical and theoretical applications.

Parallel repetition was shown to reduce the error probability of Arthur-Merlin games at an exponential rate [3] (That is,  $k$  parallel repetitions of a protocol with error  $\epsilon$  results in a protocol with error  $\epsilon^k$  for  $k \leq \text{poly}(n)$ .) It can be shown that the same is true for interactive proofs, although a formal proof does not seem to have appeared. Beyond that, parallel repetition is more problematic. In single prover proofs, Goldreich and Krawczyk [19] showed that parallel repetition does not preserve zero-knowledge. In multi-prover proof systems, whether or not parallel repetition reduces the error has been the subject of much research (see [13] for a survey). There are examples of protocols for which two parallel repetitions fail to reduce the error at all [12], so a result as strong as for the single prover model does not hold. However, the error can be reduced at an exponential rate depending on the communication complexity of the given protocol [29], and this is the best possible [16].

Soon after the appearance of interactive proofs, the notion of arguments (also called computationally convincing protocols) was put forth by [8, 9]. The difference between a “proof” and an “argument” is that the verifier in a proof is protected against false provers of unlimited computational ability, whereas for an argument, the guarantee of protection is that it is computationally infeasible for a bad prover to convince the verifier with high probability. (More precisely, the soundness condition is “computational,” holding only for polynomial time provers.) When designing cryptographic protocols it is natural to assume all parties are polynomial time, so this is a realistic model. Typically these computationally convincing protocols are designed based on complexity assumptions, so that the difficulty of convincing the verifier to accept a false claim is related to the difficulty of solving some hard underlying computational problem, like inverting a one-way function.

Many computationally convincing protocols have been designed often involving parallel repetition or some variant that preserves zero-knowledge. It seems to have been assumed that, in analogy to the interactive proof case, parallel repetition does reduce the error in computationally convincing protocols. To our knowledge, we are the first to rigorously explore the question of whether this is the case in general.

What we find is somewhat surprising: the number of rounds determines whether or not there is a general parallel reduction. While in three rounds the error does go down as expected, there are four round protocols with no error reduction at all.

## 1.1 Our results

The main result of our paper is that parallel repetition *does not* always decrease error probabilities. Assuming the existence of trapdoor permutations we first show a protocol for which the error for two parallel repetitions is basically the same as the error of the original protocol. Next we show a protocol for which the error for  $k = \text{poly}(n)$  parallel repetitions is essentially the same as the error of the original protocol.

In the last mentioned construction, the communication complexity of the original protocol depends linearly on  $k$ . Thus, these examples still hold out the possibility of a Raz-like [29] result in which the error does decrease but at a rate proportional to the communication complexity. However, we then present evidence that even this is unlikely to hold in the computational setting. We present a protocol for which there is no “black-box” error-reduction theorem, meaning that standard techniques will be unable to show any reduction in error for even an arbitrarily large polynomial number of repetitions.

These results exploit the notion and construction of non-malleable encryption schemes of [11].

We stress that this is independent of any zero-knowledge (ZK) concerns. As we indicated above, it is well known that zero-knowledge is not preserved under parallel repetition [19]. What we are saying is that even the error does not in general go down.

These results are somewhat surprising. Computationally convincing proofs have been around for a long time, and there are a large number of protocols in the literature that use parallel repetition or some variant that preserves zero knowledge. Our results say that a claim that these protocols have low error, if true, cannot rely on a general theorem but must be justified by proofs specific to the protocol at hand. For some constant round protocols, rigorous proofs of this sort have been provided [15, 5] (Note the constructions there are not exactly parallel repetition.) More often, however, either no argument, or sketchy arguments which seem implicitly to assume parallel repetition works in general, are provided.

The example protocols that establish our negative results have four rounds of interaction. This means that for protocols of four or more rounds we cannot expect a general result saying parallel repetition reduces the error. The best we could hope for is that it does for protocols of three or less rounds. To round off our negative results, we prove this matching positive result, showing that for computationally convincing protocols of three or less rounds, parallel repetition reduces the error at about the best rate we could expect: exponentially until it becomes negligible. (We cannot expect the error probability of a computationally sound proof to ever go below negligible, as this typically is as low as we assume the probability of breaking the underlying hard computational problem like factoring.) The proof exploits techniques from [25].

These results indicate that there is a fundamental difference about computational soundness and the kind of “statistical” soundness that is the property of interactive proofs (whether single or multiple prover ones) as far as composition is concerned. They also indicate one must be careful in making claims about the error of specific computationally sound protocols.

## 1.2 The bigger picture

Although the main motivation came from the area of zero-knowledge arguments, the question of reducing error in computationally sound protocols makes sense whenever one polynomial time party is going to accept or reject the other, or, even more generally, whenever a party will produce a boolean output. One natural task where this occurs is identification [14]. Suppose that one can show a protocol where an unauthorized player has probability  $\alpha$  of making the verifier accept (whereas an authorized player may know a strategy that is perfect). Here  $\alpha$  is typically non-negligible and if this is to be used for identification, then the probability of success by an unauthorized player should be very small. It is tempting to run several copies of the protocol in parallel and hope that the probability that an

unauthorized player succeeds in all of them goes down to the desired level <sup>1</sup>

Other examples might include a coin-flipping protocol. If one party can bias the coin in one repetition by only a certain amount, what can we say about the probability that a fixed  $k$ -bit string is chosen if the protocol is executed  $k$  times in parallel? This is basically the same question just view an outcome that corresponds to the target string as an “accept” and the reverse as a “reject”

### 1.3 A closer look

Let us try to give some idea of why error reduction by parallel repetition in computationally sound protocols might be problematic and what are the issues involved

The basic question of whether repetition can reduce error is a form of the *direct product* question. A direct product conjecture asks whether, for some model, and a suitably hard computational problem for that model, several independent instances of the problem are harder than a single instance. A classical example of such a result is Yao’s XOR lemma [31, 26, 20, 24, 25] which states that, if any feasible computation in a non-uniform model has a constant chance of failure at predicting a Boolean function, then the probability that a feasible computation could compute the  $\oplus$  of several strings becomes negligible. To view this problem as a direct product result becomes clearer in the two rounds case. View the verifier’s first message as a challenge or instance of a computational problem. The prover then needs to compute a string (the response) which bears a certain relationship to the challenge (the verifier accepts). Since the verifier’s messages are independent, this is a direct product question for relations. One complication is that the prover cannot necessarily compute this relation, since whether the verifier accepts depends not just on the challenge, but on a random tape used to pick the challenge. (For example, the challenge could be a one-way function of the random tape.) This correspondence is the basic idea of the positive results for three round protocols, which uses a modified version of the proof in [25] of a direct product for Boolean functions. Intuitively, the reason this proof could be adapted for three round protocols is that, although we cannot tell whether the actual verifier accepts, we are interested in converting a strategy for the prover in  $k$  parallel runs to one for a single run. In using the parallel strategy, we can simulate all but one of the parallel verifiers, picking their random tapes. We can thus use whether the simulated verifiers accept as an indicator for whether the real verifier is likely to accept. If the simulated verifiers are likely to reject then we back up and re-simulate them using fresh random tapes until most of them accept.

Unfortunately, this breaks down for the four round case. Here, there are two rounds of challenges and two rounds of responses. After the first round of challenges, the single run prover can pick a response that has a non-negligible chance of causing most of the simulated verifiers to accept. However, at this point, the prover must commit herself to the first response, and hence to the simulated verifier’s first challenges. Their second challenges could be fixed as functions of their first challenges, so in general, if the simulated prover’s response is rejected by many of the simulated verifiers there is no way to back up and try again. We formalize this by having the first challenge be an encryption of the second. It is important that this encryption be *non-malleable* ([11])<sup>2</sup>. The resulting protocol looks much like protocols obtained by a general technique to make trusted verifier zero-knowledge protocols truly zero-knowledge.

<sup>1</sup> The basic protocol is usually cryptographic, but this idea also makes sense in non-traditional situations. For instance, in [28] there is a proposal to use “an automated Turing test” to make sure that a human is requesting to use a resource like an on-line database and for combating junk-mail. The idea is that the user receives as a challenge a question (or task) that is easy for human but where computers have not made much progress (e.g. simple visual or linguistic problems). Since the probability that a computer will succeed in the task is non-negligible, a natural question is whether by repeating several task in parallel one reduces significantly the probability of success of a machine.

<sup>2</sup>It is interesting to note that while the (single-fold) protocol works due to the non-malleability of the cryptosystem the repeated protocol fails due to the malleability of the protocol itself.

## 2 Definitions and setup

If  $f_1, f_2: \Lambda \rightarrow \mathbb{R}$  are functions defined over some common domain  $\Lambda \subseteq \{0, 1\}^*$  we say that  $f_1$  is *eventually less than*  $f_2$ , written  $f_1 \leq_{\text{ev}} f_2$ , if there is an integer  $k$  such that  $f_1(\lambda) \leq f_2(\lambda)$  for all  $\lambda \in \Lambda$  with  $|\lambda| \geq k$ . We say that  $f: \Lambda \rightarrow \mathbb{R}$  is *negligible* if  $f \leq_{\text{ev}} |\cdot|^{-c}$  for every positive integer  $c$ , where  $|\cdot|^{-c}$  stands for the function  $\lambda \mapsto |\lambda|^{-c}$ . For any integer  $m$  we let  $[m] = \{1, \dots, m\}$ .

### 2.1 Computationally sound protocols

**TWO-PARTY PROTOCOLS** We consider a very general two party protocol setting. Think of the players as having some common initial context, represented by a binary string  $\lambda$ , member of an underlying set  $\Lambda \subseteq \{0, 1\}^*$  called the *domain*. (This  $\lambda$  might be, for example, messages from some previous protocol, thus possibly involving other parties, or public keys of these or other players.) The length of  $\lambda$ , denoted  $n$ , functions as the security parameter. The actual input for the protocol between the two players is a string  $x$ , drawn according to some input distribution  $I$ , namely  $x \stackrel{R}{\leftarrow} I(\lambda)$ . The first party called the *prover* is trying to convince the second party, called the *verifier*, of some claim related to  $x$ . They exchange messages, and, at the end of the interaction, the verifier either accepts (outputs 1) or rejects (outputs 0). We are mostly interested in the case where the parties run in time polynomial in  $n$ . The verifier is fixed in our setting, so that the protocol is fully specified given the strategy of the prover.

We view a party  $B$  (whether prover or verifier) as an interactive algorithm. It takes inputs  $x$ , the conversation  $M_1, \dots, M_i$  so far, and its random tape  $R$  to output the next message, denoted  $B(x, M_1, \dots, M_i, R)$ . (For simplicity we omit  $\lambda$  from the notation. It is assumed all parties always have access to this context.) In the case of the verifier the last message is identified with the bit that indicates its decision. We let  $B_x(\cdot) = B(x, \cdot, \cdot)$  denote  $B$  with input fixed to  $x$ .

**COMPUTATIONAL SOUNDNESS** Let  $A$  be any interactive algorithm playing the role of the prover. We let  $\text{Acc}(A, V, x)$  denote the probability that  $V$  accepts in its interaction with  $A$  on common input  $x$ , the probability being over the coin tosses of both parties, with  $x$  fixed. We let  $\text{Acc}(A, V, I, \lambda)$  denote the probability that  $V$  accepts in its interaction with  $A$  on common input  $x$  where the probability is over  $x$  drawn randomly from  $I(\lambda)$  and the coin tosses of both parties. We are interested in computational soundness, namely the probability that  $V$  can be made to accept, by a polynomial time prover, measured as a function of  $n$ . The error probability is given by a function  $\epsilon: \Lambda \rightarrow \mathbb{R}$ .

**Definition 2.1** Let  $V$  be a verifier strategy over a domain  $\Lambda$  and input distribution  $I$ . We say that  $V$  has (computational) error probability  $\epsilon(\cdot)$  if for every polynomial time prover  $P$  it is the case that  $\text{Acc}(P, V, I, \cdot) \leq_{\text{ev}} \epsilon(\cdot)$ .

That is, the probability that a prover can convince  $V$  to accept is at most  $\epsilon(\lambda)$  for long enough  $\lambda$ , with how long depending on the prover.

**REMARKS** As indicated above, this is a very general setup in that we allow a context and input distribution. The “arguments” model of [9, 8] is typically presented in terms of language recognition. That’s a special case of our setup. To discuss a proof system or argument for a language  $L$  let  $\Lambda = \bar{L}$  and let  $I(\lambda)$  simply assign probability one to  $\lambda$  and zero to every other string, for each  $\lambda \in \Lambda$ . Then the soundness condition of Definition 2.1 collapses to the standard one. In particular, this means all our positive results apply to the standard argument model.

In any usage, the protocol must also satisfy a completeness condition. This says that there is a particular, polynomial time prover strategy  $P$  that, if provided with some “secret” information associated to the input  $x$ , succeeds in making  $V$  accept with high probability (for example, with probability 1). This  $P$  is called the *honest prover*. We do not formally make such a condition because



the soundness, which is the main object of our study, is an independent property. But it should be understood that meaningful protocols will satisfy some form of completeness. (For example the ones in our negative results do.)

We are not discussing zero-knowledge. This may or may not be a property of our protocols. We are concerned only with soundness error.

## 2.2 Parallel repetition

Parallel repetition means the original protocol (specified by some verifier  $V$ ) is repeated *independently*  $k$  times in parallel, where  $k \leq \text{poly}(n)$ . We let  $V^k$  denote the corresponding verifier whose strategy consists of running  $k$  independent copies of  $V$ , and accepting iff  $V$  would accept in all sub-protocols. More formally, the random tape  $R$  of  $V^k$  has the form  $R_1 \dots R_k$  where each  $R_i$  is a random tape for  $V$ . The  $i$ -th message in the protocol, denoted  $M_i$ , is parsed as  $M_i = M_{i,1} \dots M_{i,k}$ . The reply  $M_{i+1} = V^k(x, M_1 \dots M_i, R)$  of  $V^k$  to some message  $M_i$  is defined via  $M_{i+1,j} = V(x, M_{1,j} \dots M_{i,j}, R_j)$  for  $j = 1, \dots, k$ . Note that the verifier  $V^k$  looks for a unanimous vote amongst the sub-protocols, accepting iff *all* of them are accepting for  $V$ .

Note that the prover need not respect the sub-protocols in his replies. That is, if the prover must compute a reply  $M_{i+1}$  to  $M_i$ , it can choose  $M_{i+1,j}$  to depend on all previous information, including values  $M_{t,l}$  where  $l \neq j$  (and  $t \leq i$ ). This is what makes parallel repetition difficult to analyze.

Fix some domain  $\Lambda$  and input distribution  $I$ . Say  $V$  has error probability  $\epsilon(\cdot)$  over  $\Lambda$  and  $I$ , as per Definition 2.1. The parallel repetition problem is: what is the error probability  $\epsilon_k$  of  $V^k$ ?

It is important for the meaningfulness of the parallel repetition problem that we deny the verifier any secret information about either the common input  $x$  or the context  $\lambda$ . See Appendix C.

## 2.3 Black-box amplification

“Black-box” error-reduction (or amplification), as we now discuss it, is a way of proving error-reduction that is interesting for two reasons. First, it yields a strong result, and thus is desirable. Second, whenever we can prove error-reduction at all, it appears we can prove black-box error-reduction. Thus it makes sense to focus on this method.

One proves amplification by reduction. Namely, given a prover  $A$ , for the protocol defined by  $V^k$ , such that  $\text{Acc}(A, V^k, x) > \epsilon$ , we construct a prover  $B$ , for the protocol defined by  $V$ , such that  $\text{Acc}(A, V^k, x) > \delta$ . (This means that if the error probability of the protocol defined by  $V$  is at most  $\delta$  then that of the protocol defined by  $V^k$  is at most  $\epsilon$ .) The natural way (it is hard to imagine an alternative) to accomplish this transformation is “black-box”. We specify an oracle machine  $S$  such that  $B = S^A$ . Namely, to define  $B$ , we need just one strategy which can call  $A$  as a subroutine.

**Definition 2.2** Let  $V$  be a verifier strategy over a domain  $\Lambda$  and input distribution  $I$ . Suppose  $\epsilon, \delta : \Lambda \rightarrow [0, 1]$ . A  $(k, \delta, \epsilon)$  black-box prover transform for  $V$  is a probabilistic, polynomial time oracle algorithm  $S$  such that for any prover  $A$  the following is true for all  $\lambda \in \Lambda$ : if  $\text{Acc}(A, V^k, I, \lambda) > \epsilon(\lambda)$  then  $\text{Acc}(S^A, V, I, \lambda) > \delta(\lambda)$ . We say that  $V$  has a  $(k, \delta, \epsilon)$ -black-box error-reduction procedure if there exists a black-box prover transform for  $V$ .

We explain what we mean by providing  $A$  to  $S$  as an oracle. The manner in which  $S$  can call the probabilistic, interactive function  $A$  is constrained. When the common input is  $x$  (a point in the support of  $I(\lambda)$ ), algorithm  $S$  has oracle access to  $A_x$ . (It cannot invoke  $A$  on common inputs other than  $x$ .) Furthermore it does not directly supply, or even have access to, the random tape to  $A_x$ . Think of a random tape  $R$  for  $A_x$  as automatically chosen at random and fixed.  $S$  can then supply conversation prefixes  $c$  and get back  $A_x(c, R)$ . (Note this means  $S$  can “back-up”  $A$  on the same random tape.) Also,  $S$  has a special “reset” button: when it hits this, a new random tape  $R$  is chosen at random and fixed for  $A_x$ .

It is important that  $S$  is polynomial time, but note that  $A$  is not restricted to be polynomial time. Of course in the computational soundness setting we are only interested in the case where  $A$  is polynomial time, but it is hard to imagine a natural black-box procedure that differentiates these cases. From the point of view of  $S$ , prover  $A$  is just an oracle to be invoked (at unit cost per oracle call) the efficiency of  $A$  doesn't matter. Of course,  $S$  can only call  $A$  a polynomial number of times.

Note that our error-reduction theorem for three round protocols (namely Theorem 4.1) indeed presents a black-box prover transform. In fact it is stronger.

### 3 Parallel repetition fails in general

In this section we provide our negative results. Proofs of all claims here can be found in Appendix A.

#### 3.1 Non-malleable encryption

Our constructions exploit non-malleable encryption schemes as defined and constructed in [11]. Let  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  specify a public key encryption scheme. The key generator algorithm  $\mathcal{G}$  takes input  $1^n$  and produces a pair  $(pk, sk)$  of matching public and secret keys. Given the public key  $pk$ , a message bit  $b$ , and coins  $r$  the encryption algorithm  $\mathcal{E}$  produces a ciphertext written  $C = \mathcal{E}_{pk}(b, r)$ . (When we write  $C \stackrel{R}{\leftarrow} \mathcal{E}_{pk}(b)$  it means the underlying choice of  $r$  is made at random.) Decryption algorithm  $\mathcal{D}$  takes the secret key and ciphertext to recover the message,  $b = \mathcal{D}_{sk}(C)$ . Note  $\mathcal{G}$  and  $\mathcal{E}$  are probabilistic while  $\mathcal{D}$  is deterministic, and all algorithms are polynomial time. We assume *unique decryptability* for any bit  $b$  and string  $r$  it is the case that  $\mathcal{D}_{sk}(\mathcal{E}_{pk}(b, r)) = b$ . This means there does not exist a triple  $C, r_0, r_1$  such that  $C = \mathcal{E}(0, r_0) = \mathcal{E}(1, r_1)$ . This will be important for us.

Suppose the adversary is given  $C \stackrel{R}{\leftarrow} \mathcal{E}_{pk}(b)$  for some random bit  $b$ . According to the standard notion of semantic security [22] she cannot figure out  $b$ . We want a stronger property, namely that she cannot modify  $C$  to some different ciphertext  $C'$  whose corresponding plaintext is related to the plaintext of  $C$ . This is not guaranteed by semantic security (and in fact for many semantically secure cryptosystems it is easy given an encryption of a bit, to create an encryption of the complement bit). But it is guaranteed by non-malleability. We do not provide a formal definition here (see [11]).

Our first protocol requires only “complement security,” meaning it is hard, given an encryption  $C$  of a bit  $b$ , to come up with an encryption  $C'$  of  $1 - b$ . Our second protocol requires “copy security,” meaning it is hard, given an encryption  $C$  of a bit  $b$ , to come up with an encryption  $C'$  of  $b$  such that  $C' \neq C$ . (Note in either case if  $C'$  is not a valid encryption of any bit, then it “counts” as a failure.) Any non-malleable scheme has these two properties. Our third protocol actually uses non-malleability in its strongest form as per [11].

It is shown in [11] that non-malleable (and hence complement and copy secure) encryption schemes with unique decryptability exist given the existence of trapdoor permutations. Accordingly we assume such a scheme  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is given and let  $\nu : \{1^n : n \in \mathbb{N}\} \rightarrow \mathbb{R}$  be a negligible function that eventually upper bounds the success of any (polynomial time) adversary. We call this function the *security* of the encryption scheme. See Appendix A for a more formal development.

#### 3.2 Two fold parallel repetition fails

We specify a four round protocol that has error about  $1/2$ , but when repeated twice in parallel the error is still about  $1/2$ , rather than  $1/4$ .

The input to the parties is a public key  $pk$  of the above encryption scheme. The prover is claiming that he “knows” the decryption key  $sk$ , or, more exactly, that he knows how to decrypt. The verifier  $V$  wants to test this. Roughly, the idea is that  $V$  sends a ciphertext  $B \stackrel{R}{\leftarrow} \mathcal{E}_{pk}(b)$  for a random bit  $b$ , and the prover must succeed in returning an encryption of the bit  $b$  complemented. The ability to do

this is viewed as corresponding to the ability to decrypt  $B$ . The full specification of the protocol is below

In specifying the protocol we give the instructions for  $V$  and also indicate what kinds of messages the prover is providing. But we specify no particular strategy for the prover: this is irrelevant to the analysis since we are interested only in soundness. The name of the protocol is  $DD_2$ , standing for “Don’t Do twice”

**Protocol  $DD_2$**

**Common input:**  $pk$

- (1)  $V$  picks a random bit  $b \in \{0, 1\}$  and coins  $r$ . It sets  $B = \mathcal{E}_{pk}(b, r)$  and sends  $B$  to the prover
- (2) The prover sends a ciphertext  $C$  to  $V$
- (3)  $V$  sends  $b, r$  to the prover
- (4) The prover sends a bit  $c$  and a string  $s$  to  $V$
- (5)  $V$  accepts iff  $\mathcal{E}_{pk}(c, s) = C$  and  $b \neq c$

To complete the protocol specification we still have to ask questions like: where does the key  $pk$  come from? Formally, we fit the protocol into the framework of Section 2.1 by considering the context to be the security parameter,  $\lambda = 1^n$ , so that the domain is  $\Lambda = \{1^n \mid n \geq 1\}$ . The input distribution algorithm  $I$  takes  $1^n$ , runs  $\mathcal{G}(1^n)$  to get back  $(pk, sk)$ , and outputs  $pk$ . (Intuitively, this operation is performed by the honest prover who keeps  $sk$ . The dishonest prover, who is our concern for the soundness, does not know  $sk$ .) Now, for any polynomial time prover  $P$ , we can consider the acceptance probability  $\text{Acc}(P, V, I, 1^n)$ .

It is easy to see that there is a (polynomial time) strategy for the prover to make  $V$  accept with probability  $1/2$ . In Step (2) it just picks a bit  $c$  at random, picks coins  $s$  at random, sets  $C = \mathcal{E}_{pk}(c, s)$ , and sends  $C$  to  $V$ . In Step (4) it sends  $c, s$ . It wins if  $b = c$  which happens half the time. It turns out it is not possible to do much better.

**Claim 3.1** *If the encryption scheme  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is non-malleable, then the error probability of protocol  $DD_2$  is  $1/2 + \nu(\cdot)$*

Intuitively this is true since in order to win the prover must create  $C$  to be an encryption of the complement of  $b$ , given  $B \stackrel{R}{\leftarrow} \mathcal{E}_{pk}(b)$ , and this is hard if the encryption scheme is complement secure. Indeed, given a prover with probability of success  $\alpha$ , we can turn it into an adversary  $A$  that complements and has advantage at least  $\alpha - 1/2$ .

Now consider  $DD_2^2$ , the two-fold parallel repetition of  $DD_2$ . We claim its error probability is not less than  $1/2$  (let alone being about  $1/4$  as one may have wanted).

**Claim 3.2** *In the protocol consisting of two parallel repetitions of  $DD_2$ , there is a polynomial time strategy for the prover to make the verifier accept with probability at least  $1/2$*

### 3.3 Many-fold parallel repetition fails

We now generalize the protocol of the previous section to  $k$  repetitions. We show that for any  $k$  there exists a protocol  $DD_k$  which is a four round protocol that has error probability of about  $1/2$ , but when repeated  $k$  times in parallel the error probability does not reduce. As in  $DD_2$ , the input to the parties is a public key  $pk$  of the above encryption scheme. The prover is claiming that he “knows” the decryption key  $sk$ , or, more exactly, that he knows how to decrypt.

**Protocol  $DD_k$**

**Common input:**  $pk$

- (1)  $V$  picks a random bit  $b \in \{0, 1\}$  and coins  $r$ . It sets  $C = \mathcal{E}_{pk}(b, r)$  and sends  $C$  to the prover
- (2) The prover sends  $k - 1$  (single-bit) ciphertexts  $C_1, C_2, \dots, C_{k-1}$
- (3)  $V$  sends  $b, r$  to the prover
- (4) The prover sends  $k - 1$  pairs of (bit, random-string)  $(c_1, s_1), (c_2, s_2), \dots, (c_{k-1}, s_{k-1})$  to  $V$
- (5)  $V$  accepts iff
  - For all  $1 \leq i \leq k - 1$   $\mathcal{E}_{pk}(c_i, s_i) = C_i$
  - $B \notin \{C_1, C_2, \dots, C_{k-1}\}$
  - $\bigoplus_{i=1}^{k-1} c_i \neq b$

Using similar argument to the proof of Claim 3.1 we can show

**Claim 3.3** *If the encryption scheme  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is non-malleable, then the error probability of protocol  $\text{DD}_k$  is  $1/2 + \nu(\cdot)$*

However, we can also claim that if protocol  $\text{DD}_k$  is repeated  $k$  times in parallel the error probability does not reduce significantly

**Claim 3.4** *In the protocol consisting of  $k$  parallel repetitions of  $\text{DD}_k$ , there is a polynomial time strategy for the prover to make the verifier accept with probability at least  $1/2$*

### 3.4 Failure of parallel error reduction with low communication

In light of the results of Raz [29] and Feige and Verbitsky [16] a reasonable conjecture at this point is that the failure of error-reduction in protocol  $\text{DD}_k$  is due to the fact that the communication complexity is proportional to  $k$  (the number times we are going to execute the protocol in parallel). In other words, the above results still leave open the possibility that for any protocol there is a value  $\alpha > 0$  such that if the protocol is repeated  $k$  times in parallel, then the probability of failure in all  $k$  execution is  $\alpha^{-k}$ , for sufficiently large  $k$  (the value of  $\alpha$  is determined by the amount of communication in the protocol). This is the case with two-prover proof systems.

However, we now give strong indication that for computationally sound protocols even this is *not* the case. We show (assuming non-malleable cryptosystems exist) that there is no general black-box error-reduction procedure, where black-box means that one does not look inside computation of the players, but can just execute them and watch their behavior, as defined in Definition 2.2. We do this by presenting a particular protocol LC (“low communication”) for which we can prove (assuming non-malleable cryptosystems exist) that there is no black-box error-reduction procedure.

The common input in protocol LC consists of a pair  $(pk_1, pk_2)$  of public keys drawn independently at random according to  $\mathcal{G}$ . Thus, formally, the domain is again  $\Lambda = \{1^n \mid n \in \mathbb{N}\}$  and the input distribution  $I$  is the function which on input  $1^n$  outputs  $(pk_1, pk_2)$  where  $pk_1 \xleftarrow{R} \mathcal{G}(1^n)$  and  $pk_2 \xleftarrow{R} \mathcal{G}(1^n)$ .

#### Protocol LC

**Common input:**  $pk_1, pk_2$

- (1)  $V$  picks a random trit  $b \in \{0, 1, 2\}$  and coins  $r$ . It sets  $B = \mathcal{E}_{pk_1}(b, r)$  and sends  $B$  to the prover
- (2) The prover sends a ciphertext  $C$  to  $V$
- (3)  $V$  sends  $b, r$  to the prover
- (4) The prover sends a trit  $c \in \{0, 1, 2\}$  and a string  $s$  to  $V$
- (5)  $V$  accepts iff  $\mathcal{E}_{pk_2}(c, s) = C$  and  $b + c = 0 \pmod 3$



The following theorem says there is no black-box prover transform to show that the error of LC even reduces a little under lots of repetitions (It is understood we mean over input distribution  $I$  defined above)

**Theorem 3.5** Assume  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is a non-malleable encryption scheme with security a negligible function. Let  $k = k(n)$  be any polynomial. Let  $\delta > 1/3$  be a constant and let  $\epsilon : \Lambda \rightarrow [0, 1]$  be arbitrary. Then there is no  $(k, \epsilon, \delta)$ -black-box error-reduction procedure for LC.

(Actually it is enough that  $\delta(\cdot) \geq 1/3 + \tau(\cdot)$  where  $\tau$  is a non-negligible function.) Note the communication complexity of LC does not depend on  $k$ , unlike  $DD_k$ .

Theorem 3.5 clearly follows from Claim 3.6 below, which says that there is a prover strategy  $F$  for  $LC^k$  which succeeds in convincing  $V^k$  with probability about  $1/3$ , no matter how large is  $k$ . However, given this strategy as an oracle, there is no way to convince the original verifier with probability significantly more than  $1/3$ .

**Claim 3.6** Assume  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is a non-malleable encryption scheme with security a negligible function. Let  $k = k(n)$  be any polynomial. There is a prover  $F$  for protocol  $LC^k$  and a negligible function  $\nu(\cdot)$  such that

- (1)  $\text{Acc}(F, V^k, I, 1^n) \geq 1/3 - \nu(1^n)$  for all  $n \in \mathbb{N}$
- (2)  $\text{Acc}(S^F, V, I, \cdot) \leq_{\text{ev}} 1/3 + \nu(\cdot)$  for any polynomial-time oracle algorithm  $S$

The prover  $F$  will not be polynomial time. But that's permitted by Definition 2.2 and we explained there the rationale for this decision.

In fact we show something stronger. We will now define a certain “oracle”  $\mathcal{O}_{pk_1, pk_2}(\cdot)$  and make some claims about the protocol when parties have access to this oracle. From this we will deduce Claim 3.6. The oracle is, intuitively, like a prover for  $LC^m$ . (For our purposes it suffices to set  $m = k - 1$ .) It has two “stages,” and maintains state between invocations of the two stages. In the first stage it takes as input  $m$  ciphertexts  $C_1, \dots, C_m$  under key  $pk_1$ .

Oracle  $\mathcal{O}_{pk_1, pk_2}((C_1, \dots, C_m))$

- (1) If any of  $C_1, \dots, C_m$  is invalid (meaning not the encryption of any bit under  $pk_1$ ) then reject. (The ciphertexts in the cryptosystem of [11] are self-validating, so this step does not reveal any extra information to the caller of the oracle.) If not, we know there are bits  $x_1, \dots, x_m$  and strings  $t_1, \dots, t_m$  such that  $C_i = \mathcal{E}_{pk_1}(x_i, t_i)$  for  $i = 1, \dots, m$ .
- (2) If there is some  $i \neq j$  such that  $C_i = C_j$  then again reject.
- (3) Else, decrypt the ciphertexts to get the plaintexts  $x_1, \dots, x_m$ . Set  $x = x_1 + \dots + x_m \bmod 3$ .
- (4) Pick  $s$  at random and return  $C = \mathcal{E}_{pk_2}(x, s)$ .
- (5) Store a ‘context’ for this invocation, including the information  $x, s$ .

Having been invoked on  $C_1, \dots, C_m$  and having returned  $C$  as above, the oracle can be invoked for a continuation of the interaction. Here, it is fed  $(b_1, r_1), \dots, (b_m, r_m)$ .

Oracle  $\mathcal{O}_{pk_1, pk_2}((C_1, \dots, C_m), ((b_1, r_1), \dots, (b_m, r_m)))$

- (1) Check that  $\mathcal{E}_{pk_1}(b_i, r_i) = C_i$  for all  $i \in [m]$  and if this is not true then reject.
- (2) Return  $(x, s)$  where these quantities are as computed and stored, in the code for the first stage above.

Note that  $\mathcal{O}$  is *not* computable in polynomial time, since (in its first stage) it decrypts, without having access to the decryption key.

Also  $\mathcal{O}$  is not an “oracle” in the traditional sense since it maintains state between invocations. Since  $\mathcal{O}$  is just a tool in proving Claim 3.6 this doesn't matter much, but in any case we note that

this state can be eliminated by specifying  $s$  as  $F_K((C_1, \dots, C_m))$  where  $F$  is a pseudorandom function family [18] and the key  $K$  is chosen at random and made a part of the description of  $\mathcal{O}$

We first claim that given access to this oracle, it is possible to make the verifier  $V^k$  of  $LC^k$  accept  $1/3$  of the time

**Claim 3.7** *There is a polynomial time oracle algorithm  $M$  and a negligible function  $\nu(\cdot)$  such that  $\text{Acc}(M^{\mathcal{O}_{pk_1, pk_2}}, V^k, (pk_1, pk_2)) \geq 1/3 - \nu(1^n)$  for any input  $(pk_1, pk_2)$  to  $LC^k$*

On the other hand, the non-malleability of the underlying cryptosystem is strong enough to ensure that access to this oracle does not help a polynomial time prover to convince the verifier of the original protocol with probability significantly above  $1/3$ . This is (clearly) a consequence of the following

**Claim 3.8** *Assume that the encryption scheme  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is non-malleable and  $A$  is a probabilistic polynomial time oracle algorithm that is given  $B \xleftarrow{R} \mathcal{E}_{pk_1}(b)$  where  $b \xleftarrow{R} \{0, 1, 2\}$ , and access to an oracle as described above. The probability that  $A$  succeeds in coming up with  $C$  such that  $C = \mathcal{E}_{pk_2}(-b, s)$  (for some  $s$ ) is bounded by  $1/3 + \nu(\cdot)$  for a negligible function  $\nu$*

**Proof of Claim 3.6:** Just set  $F$  to the prover  $M^{\mathcal{O}_{pk_1, pk_2}}$  where  $M$  is as in Claim 3.7. The latter claim implies part (1) of Claim 3.6. Now we argue part (2) of Claim 3.6. We said above that (as a consequence of Claim 3.8), oracle access to  $\mathcal{O}_{pk_1, pk_2}$  won't help a polynomial time prover convince the verifier of  $LC$  with probability significantly above  $1/3$ . Then oracle access to  $F$  certainly won't help do it, since  $F$  can be implemented in polynomial time with access to  $\mathcal{O}_{pk_1, pk_2}$ . ■

As already indicated, Theorem 3.5 follows. It remains to show the last two claims. Their proofs are in Appendix A.

### 3.5 Extensions

We could use non-malleable bit commitment instead of non-malleable encryption in some of these protocols to get the same result. This might lead to reducing the complexity assumptions under which the result is obtained. However, the current protocol is more natural. Also using the currently best known schemes for bit commitment would have increased the number of rounds of the protocols.

We can get similar results for protocols for proving membership in an NP language, like this. Let  $L$  be any such language. Let the input be  $x, y$  where  $y$  is an input for one of the above protocols. (That is, a public key for a non-malleable encryption scheme in the first two protocols, and two such keys for the second.) The domain is  $\Lambda = \overline{L} \times \{1^n \mid n \geq 1\}$  and the input distribution puts on  $y$  the probability as needed by our protocols above. Run some standard argument protocol on input  $x$  and then (after this protocol has completed) run one of our protocols on input  $y$ . Accept iff both sub-protocols accept. This protocol is a proof of membership in  $L$  but the error does not reduce under parallel repetition.

## 4 Parallel repetition reduces the error of three round protocols

In this section, we show that parallel repetition does indeed decrease error probabilities for any three pass protocol where the verifier has no secret input. The technique used is based on the XOR Casino game of [25].

Let  $V$  be a verifier defining a three message protocol. Thus  $V$ 's output is either 1 (accept) or 0 (reject). Say  $V$ 's random tape is of length  $r$ . Let  $k$  be a positive integer. The following theorem states a very general error-decreasing property for three-round protocols. We let  $\text{Comm}(A, B, x)$  be the communication complexity of the interaction between parties  $A, B$  on an input  $x$ .

**Theorem 4.1** Suppose  $0 < \epsilon, \delta < 1$  and  $k \geq 2$  is an integer. Suppose  $\epsilon > (16/\delta) e^{-\delta^2 k/128}$ . Then there is an oracle algorithm  $S$  such that for any prover  $P^*$ , verifier  $V$  and input string  $x$ , the following is true. If  $\text{Acc}(P^*, V^k, x) \geq 2\epsilon$  then  $\text{Acc}(S^{P^*, V}, V, x) \geq 1 - \delta$ . Furthermore,  $S^{P^*, V_x}$  runs in time  $\text{poly}(k, |x|, 1/\epsilon, \text{Comm}(P^*, V^k, x))$ .

Here  $S$  requires only oracle access to  $P^*$ . But in fact  $S$  does not depend on  $V$  either, in the sense that oracle access to  $V$  will suffice too. A consequence of this is

**Corollary 4.2** Let  $V$  be a three round verifier strategy over a domain  $\Lambda$  and input distribution  $I$ , and let  $k = k(n)$  be  $O(\log n)$ . Then  $V$  has a  $(k, \epsilon, 1 - \delta)$ -black-box error-reduction procedure for any  $\epsilon, \delta : \Lambda \rightarrow (0, 1)$  satisfying  $\epsilon > (32/\delta) e^{-\delta^2 k/128}$ .

In terms of error probabilities, this implies the error decreases at an exponential rate.

**Corollary 4.3** Let  $V$  be a three round verifier strategy over a domain  $\Lambda$  and input distribution  $I$  with error probability  $1 - \delta$ , and let  $k = k(n)$  be  $O(\log n)$ . Then  $V^k$  has error probability  $\epsilon$  where  $\epsilon(\cdot) = (32/\delta(\cdot)) e^{-\delta(\cdot)^2 k/128}$ .

Note the error goes down at an exponential rate but only to  $1/\text{poly}(n)$ . Since the running time of  $S$  in Theorem 4.1 is a polynomial in  $1/\epsilon$ , and this running time must stay polynomial, we can only allow  $k$  to go as low as  $O(\log n)$ , which means  $\epsilon$  is  $1/\text{poly}(n)$ .

## 5 Open problems and on-going work

Can one show a positive result (ie. that parallel repetition reduces the error) for Arthur-Merlin games of more than three rounds? As a first step one might consider any constant number of rounds, and then more.

Our results are about soundness. What about proofs of knowledge [4]? Bellare, Halevi and Naor are investigating this question. By using the protocols here they have similar negative results for proofs of knowledge.

## 6 Acknowledgments

We thank Oded Goldreich for helpful comments on an earlier version of this paper.

## References

- [1] S. ARORA AND C. LUND. Hardness of Approximations. In *Approximation algorithms for NP-hard problems*, edited by DORIT HOCHBAUM, PWS Publishing Company, Boston, 1997.
- [2] L. BABAI. Trading Group Theory for Randomness. *Proceedings of the 17th Annual Symposium on the Theory of Computing*, ACM, 1985.
- [3] L. BABAI AND S. MORAN. Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes. *J. Computer and System Sciences* Vol. 36, 254-276, 1988.
- [4] M. BELLARE AND O. GOLDBREICH. On Defining Proofs of Knowledge. *Advances in Cryptology - Crypto 92 Proceedings*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992.
- [5] M. BELLARE, M. JAKOBSSON AND M. YUNG. Round-optimal zero-knowledge arguments based on any one-way function. *Advances in Cryptology - Eurocrypt 97 Proceedings*, Lecture Notes in Computer Science Vol. 1233, W. Fumy ed., Springer-Verlag, 1997.

- [6] M. BEN-OR, S. GOLDWASSER, J. KILIAN AND A. WIGDERSON Multi-Prover interactive proofs: How to remove intractability assumptions. *Proceedings of the 20th Annual Symposium on the Theory of Computing*, ACM, 1988
- [7] M. BLUM Coin Flipping over the Telephone. *IEEE COMPCON* 1982
- [8] G. BRASSARD AND C. CRÉPEAU Non-transitive Transfer of Confidence: A perfect Zero-knowledge Interactive protocol for SAT and Beyond. *Proceedings of the 27th Symposium on Foundations of Computer Science*, IEEE, 1986
- [9] G. BRASSARD, D. CHAUM AND C. CRÉPEAU Minimum Disclosure Proofs of Knowledge. *J. Computer and System Sciences*, Vol. 37, 1988, pp. 156-189
- [10] G. BRASSARD, C. CRÉPEAU AND M. YUNG Constant round perfect zero knowledge computationally convincing protocols. *Theoretical Computer Science*, Vol. 84, No. 1, 1991
- [11] D. DOLEV, C. DWORK AND M. NAOR Non-malleable cryptography. *Proceedings of the 23rd Annual Symposium on the Theory of Computing*, ACM, 1991. Full version available from authors
- [12] U. FEIGE On the success probability of two provers in one round proof systems. *Proceedings of the 6th Annual Conference on Structure in Complexity Theory*, IEEE, 1991
- [13] U. FEIGE Error reduction by parallel repetition - the state of the art, Technical Report CS95-32, Weizmann Institute
- [14] U. FEIGE, A. FIAT, AND A. SHAMIR Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, Vol. 1, 1988, pp. 77-94
- [15] U. FEIGE AND A. SHAMIR Zero-knowledge proofs of knowledge in two rounds. *Advances in Cryptology - Crypto 89 Proceedings*, Lecture Notes in Computer Science Vol. 435, G. Brassard ed., Springer-Verlag, 1989
- [16] U. FEIGE AND O. VERBITSKY Error reduction by parallel repetition- A negative result. *Proceedings of the 11th Annual Conference on Structure in Complexity Theory*, IEEE, 1996
- [17] O. GOLDBREICH Foundations of cryptography: Fragments of a book. Weizmann Institute of Science, February 1995
- [18] O. GOLDBREICH, S. GOLDWASSER AND S. MICALI How to construct random functions. *Journal of the ACM*, Vol. 33, No. 4, 1986, pp. 210-217
- [19] O. GOLDBREICH AND H. KRAWCZYK On the Composition of Zero Knowledge Proof Systems. *SIAM J. on Computing*, Vol. 25, No. 1, pp. 169-192, 1996
- [20] O. GOLDBREICH, N. NISAN AND A. WIGDERSON On Yao's XOR lemma. *Electronic Colloquium on Computational Complexity*, TR95-050, 1995
- [21] O. GOLDBREICH AND Y. OREN Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, Vol. 7, No. 1, 1994, pp. 1-32
- [22] S. GOLDWASSER AND S. MICALI Probabilistic Encryption. *J. Computer and System Sciences*, Vol. 28, 1984, pp. 270-299
- [23] S. GOLDWASSER, S. MICALI AND C. RACKOFF The knowledge complexity of interactive proof systems. *SIAM J. on Computing*, Vol. 18, No. 1, pp. 186-208, February 1989
- [24] R. IMPAGLIAZZO Hard-core distributions for somewhat hard problems. *Proceedings of the 36th Symposium on Foundations of Computer Science*, IEEE, 1995
- [25] R. IMPAGLIAZZO AND A. WIGDERSON  $P = BPP$  unless  $E$  has sub-exponential circuits. *Proceedings of the 29th Annual Symposium on the Theory of Computing*, ACM, 1997
- [26] L. LEVIN One-way functions and pseudorandom generators. *Combinatorica*, Vol. 7, No. 4, 1987, pp. 357-363

- [27] S. MICALI, CS proofs *Proceedings of the 35th Symposium on Foundations of Computer Science*, IEEE, 1994
- [28] M. NAOR, Verification of a human in the loop or Identification via the Turing Test *Manuscript*, 1996
- [29] R. RAZ, A parallel repetition theorem *Proceedings of the 27th Annual Symposium on the Theory of Computing*, ACM, 1995
- [30] M. TOMPA AND H. WOLL, Random Self-Reducibility and Zero-Knowledge Proofs of Possession of Information *Proceedings of the 28th Symposium on Foundations of Computer Science*, IEEE, 1987
- [31] A. C. YAO, Theory and Applications of Trapdoor functions *Proceedings of the 23rd Symposium on Foundations of Computer Science*, IEEE, 1982

## A Proofs for Section 3

Let's formalize complement and copy security, beginning with the former. Let  $A$  be an adversary that takes  $pk, C$ , where  $C = \mathcal{E}_{pk}(b)$ , and outputs  $C'$ . We say  $A$  is successful if  $\mathcal{D}_{sk}(C') = 1 - b$  and let

$$\begin{aligned} \text{CompSucc}_A(1^n) = & \Pr \left[ \mathcal{D}_{sk}(C') = 1 - b \mid (pk, sk) \xleftarrow{R} \mathcal{G}(1^n), b \xleftarrow{R} \{0, 1\}, \right. \\ & \left. C \xleftarrow{R} \mathcal{E}_{pk}(b), C' \xleftarrow{R} A(pk, C) \right] \end{aligned}$$

Of course,  $A$  can be successful half the time by guessing  $b$  and encrypting its complement.  $A$ 's *advantage* is the excess of its success probability over one-half. Set  $\text{CompAdv}_A(1^n) = \text{CompSucc}_A(1^n) - 1/2$ .

As for copy security, we should first rule out the case of exact copying, which is of course impossible to prevent. Formally, let  $A$  be an adversary that takes  $pk, C$  and tries to output  $C'$ . We say  $A$  is successful if  $\mathcal{D}_{sk}(C') = b$  and  $C' \neq C$ . We let

$$\begin{aligned} \text{CopySucc}_A(1^n) = & \Pr \left[ \mathcal{D}_{sk}(C') = b \text{ and } C' \neq C \mid (pk, sk) \xleftarrow{R} \mathcal{G}(1^n), b \xleftarrow{R} \{0, 1\}, \right. \\ & \left. C \xleftarrow{R} \mathcal{E}_{pk}(b), C' \xleftarrow{R} A(pk, C) \right] \end{aligned}$$

and  $\text{CopyAdv}_A(1^n) = \text{CopySucc}_A(1^n) - 1/2$ .

**Definition A.1** Encryption scheme  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is complement secure with failure probability  $\nu(\cdot)$  if for every polynomial time adversary  $A$  it is the case that  $\text{CompAdv}_A(\cdot) \leq_{\text{ev}} \nu(\cdot)$ . Encryption scheme  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is copy secure with failure probability  $\nu(\cdot)$  if for every polynomial time adversary  $A$  it is the case that  $\text{CopyAdv}_A(\cdot) \leq_{\text{ev}} \nu(\cdot)$ .

From [11] we have

**Claim A.2** If an encryption scheme  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$  is non-malleable, then it is complement secure and copy secure with negligible failure probability  $\nu(\cdot)$ .

**Proof of Claim 3.1:** We have to show that Definition 2.1 is met with  $\epsilon(\cdot) = 1/2 + \nu(\cdot)$ . Assume not. So there is some polynomial time prover  $P$  such that  $\text{Acc}(P, V, I, 1^n) > 1/2 + \nu(1^n)$  for all  $n \in N$ , where  $N$  is some infinite set of integers. We claim this contradicts the complement-security of the underlying encryption scheme  $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ . To show this we define the following adversary  $A$ . It receives input  $pk, B$ . It thinks of  $B$  as the first message from  $V$ , and computes  $C = P(pk, C, R)$ , the response that  $P$  would give, where  $R$  is a random string used as the coins  $A$  choses for  $P$ . Now it outputs  $C$ .

We claim that  $\text{CompSucc}_A(1^n) \geq \text{Acc}(P, V, I, 1^n)$  for all  $n$ . Note there is something interesting in this claim made at this point, since  $A$  has not fully simulated the interaction between  $P$  and  $V$  to the

point of seeing whether the latter accepts. Indeed,  $A$  cannot expect to do that, since in the next step  $V$  provides the decryption of  $B$  and  $A$  does not know this. Nonetheless, we want to claim that if the interaction had continued,  $V$  would indeed have accepted.

If  $P$  were able to make  $V$  accept, it would have been by providing a bit  $c$  and coins  $s$  such that  $\mathcal{E}_{pk}(c, s) = C$ . Now let  $d = \mathcal{D}_{sk}(C)$ . This is not something  $A$  can compute, but we can think about it in the proof. The unique decryptability property that we have assumed on the encryption scheme means that if  $V$  accepts, it must be that  $d = c$ , and hence that  $c = 1 - b$ . But  $A$  is successful exactly when  $c = 1 - b$ . So  $A$  is successful whenever  $V$  would have accepted, meaning  $\text{CompSucc}_A(1^n) \geq \text{Acc}(P, V, 1^n)$ .

Hence  $\text{CompAdv}_A(1^n) = \text{CompSucc}_A(1^n) - 1/2 > \nu(1^n)$  for all  $n \in N$ . But this contradicts the assumption that the encryption scheme is complement secure with failure probability  $\nu(\cdot)$ . ■

**Proof of Claim 3.2:** We are running  $\text{DD}_2$  twice in parallel. The notation we use is to superscript the quantities in  $\text{DD}_2$  with the index (1 or 2) of the run. We now describe the strategy of the clever prover.

The prover receives  $B^1, B^2$  from the verifier. It computes its return ciphertexts by “crossing” these. Namely set  $C^1 = B^2$  and  $C^2 = B^1$ , and return  $C^1, C^2$ . Next it receives  $(b^1, r^1), (b^2, r^2)$  from the verifier. Again it crosses, setting  $(c^1, s^1) = (b^2, r^2)$  and  $(c^2, s^2) = (b^1, r^1)$ , and sends  $(c^1, s^1), (c^2, s^2)$  to the verifier.

The verifier accepts if  $c^1 \neq b^1$  and  $c^2 \neq b^2$ . (Also it must check that the coin tosses provided by the prover are consistent with the bits, but it is clear this works out, so we skip it.) With the responses defined as above this amounts to accepting if  $b^2 \neq b^1$  which happens with probability exactly  $1/2$  since  $b^1$  and  $b^2$  were chosen independently at random by the verifier. ■

**Proof of Claim 3.4:** We are running  $\text{DD}_k$   $k$  times in parallel. The notation we use is to superscript the variables in  $\text{DD}_k$  with the index  $1, \dots, k$  of the run. We now describe the strategy of the clever prover. We assume that none of  $B^1, B^2, \dots, B^{k-1}$  are equal<sup>3</sup>. When the verifier sends  $B^1, B^2, \dots, B^{k-1}$ , the prover sends for the  $i$ th game

$$(C_1^i, C_2^i, \dots, C_{k-1}^i) = (B^1, \dots, B^{i-1}, B^{i+1}, \dots, B^k)$$

After the verifier sends in Step (3) the pairs  $(b^1, r^1), (b^2, r^2), \dots, (b^k, r^k)$ , the Prover can open up correctly all the ciphertexts in Step (4).

It remains to see when the verifier accepts. The verifier accepts in all the  $k$  games iff  $\bigoplus_{i=1}^{k-1} b^i = 1$ . The probability that this happens is  $1/2$  therefore this is the probability of the strategy’s success. ■

**Proof of Claim 3.7:** The prover strategy  $M^{\mathcal{O}_{pk_1, pk_2}}$  is as follows

- (1) Receives ciphertexts  $B^1, \dots, B^k$  from  $V^k$  where  $B^i = \mathcal{E}_{pk_1}(b^i, r^i)$  for  $i = 1, \dots, k$
- (2) For each  $i \in [k]$  set  $C^i \leftarrow \mathcal{O}_{pk_1, pk_2}((B^1, \dots, B^{i-1}, B^{i+1}, \dots, B^k))$ . Send  $C^1, \dots, C^k$  to  $V^k$
- (3) Get back  $(b^1, r^1), \dots, (b^k, r^k)$  from  $V^k$
- (4) For each  $i \in [k]$  set  $(c^i, s^i)$  to  $\mathcal{O}_{pk_1, pk_2}((B^1, \dots, B^{i-1}, B^{i+1}, \dots, B^k), ((b^1, r^1), \dots, (b^{i-1}, r^{i-1}), (b^{i+1}, r^{i+1}), \dots, (b^k, r^k)))$   
Send  $(c^1, s^1), \dots, (c^k, s^k)$  to  $V^k$

We claim that the probability this strategy convinces  $V^k$  to accept is at least  $1/3 - \nu(\cdot)$  for some negligible function  $\nu(\cdot)$ . To see this, assume  $b^1 + \dots + b^k \equiv 0 \pmod 3$ . This happens with probability

<sup>3</sup> The probability of this event is low, being bounded by  $k^2\nu(\cdot)$ . However, even this negligible failure probability can be eliminated, we defer the discussion.



1/3 The chance that two ciphertexts are equal is negligible, so the oracle returns answers Now note that for any  $i \in [k]$  it is the case that

$$b^i + c^i \bmod 3 = b^i + (b^1 + \dots + b^{i-1} + b^{i+1} + \dots + b^k) \bmod 3 = 0 \bmod 3,$$

using the definition of the oracle Thus  $V^k$  accepts by definition of  $LC^k$  ■

**Proof of Claim 3.8:** We use here the fact that non-malleable encryption is immune against very powerful attacks the adversary is given a challenge ciphertext and a sequence of additional ciphertexts and should come up with a ciphertext whose corresponding plaintext satisfies some relation  $\mathcal{R}$  with the challenge plaintext and the plaintexts of the ciphertext sequence Also the adversary is be given access to the decryption box *after* it receives the challenge and can query it on any ciphertext of its choice except the challenge ciphertext and sequence. Still, it must fail

For our context, consider specifically an algorithm  $A'$  that receives, in addition to  $B \xleftarrow{R} \mathcal{E}_{pk_1}(b)$ , where  $b \xleftarrow{R} \{0, 1, 2\}$ , also  $3t$  ciphertexts

$$B_0^1, B_0^2, \dots, B_0^t, B_1^1, B_1^2, \dots, B_1^t, B_2^1, B_2^2, \dots, B_2^t,$$

such that  $B_j^i \xleftarrow{R} \mathcal{E}_{pk_2}(b + j \bmod 3)$  for  $j = 0, 1, 2$  and  $i \in [t]$ , where  $t$  is an upper bound on the number of oracle queries made by our given adversary  $A$  (Note the additional ciphertexts are under  $pk_2$ , not  $pk_1$ ) It wins if it comes up with a valid encryption, under  $pk_2$ , of  $-b \bmod 3$  Also  $A'$  has access to  $\mathcal{D}_{sk_1}(\cdot)$  but is not allowed to invoke it on ciphertext  $B$  The notions and constructions of non-malleable cryptosystems from [11] imply that such an adversary  $A'$  has only a  $1/3 + \nu(\cdot)$  chance of winning where  $\nu(\cdot)$  is a negligible function

Let  $\mathcal{B} = \{B_i^j : i \in [k] \text{ and } j = 0, 1, 2\}$  Now, we show that if  $A$  exists we can construct an  $A'$  as above that has a chance of winning larger than possible, thus contradicting the assumed security of the non-malleable cryptosystem  $A'$  will run  $A$  and itself provide answers to  $A$ 's oracle queries  $A'$  gives  $B$  as input to  $A$   $A'$  maintains counters  $f_0, f_1, f_2$ , all initially zero

When  $A$  asks its oracle a query  $(C_1, \dots, C_m)$ , our algorithm  $A'$  uses its decryption box to get  $c_i \leftarrow \mathcal{D}_{sk_1}(C_i)$  for any  $i \in [m]$  such that  $C_i \neq B$  Let  $j$  be the sum modulo 3 of the  $c_i$ 's, taken over all  $i$  for which  $C_i \neq B$  We now consider two cases, that  $B$  is one of  $C_1, \dots, C_k$  and that it is not In case none of  $C_1, \dots, C_m$  was equal to  $B$ , algorithm  $A'$  picks  $s$  at random and returns  $\mathcal{E}_{pk_2}(j, s)$  as the answer to the oracle query In case there was an  $i$  such that  $C_i = B$  (note we can assume  $i$  is unique because otherwise the oracle rejects anyway and thus is easy to simulate),  $A'$  increments  $f_j$  by one and returns  $B_j^{f_j}$  One can check that the distribution of these answers to oracle queries provided by  $A'$  is identical to that provided by the real oracle

Next we must consider a second stage oracle query of the form  $(C_1, \dots, C_m), ((b_1, r_1), \dots, (b_m, r_m))$   $A'$  checks that indeed  $\mathcal{E}_{pk_1}(b_i, r_i) = C_i$  for each  $i \in [m]$  and returns a reject otherwise, just as  $\mathcal{O}_{pk_1, pk_2}$  would do Now, assuming the check passes, there are two cases just as above, namely is  $B \in \{C_1, \dots, C_m\}$  or not If not,  $A'$  can easily return  $(j, s)$  where these were the quantities it chose in its first stage On the other hand if  $B = C_i$  for some  $i \in [m]$  then  $A'$  has found  $b$ , because  $b = b_i$  and  $b_i$  was provided by  $A$  Thus  $A'$  wins directly, because it can just encrypt  $-b \bmod 3$  under  $pk_2$  and output the result On the other hand if  $A'$  doesn't directly win it is simulating the oracle correctly

Finally  $A$  returns, by assumption, a valid encryption, under  $pk_2$ , of  $-b \bmod 3$ , and  $A'$  can just output this ■

## B Proof of Theorem 4.1

In the given protocol (namely that defined by  $V$ ) we can assume that the prover makes the first move, otherwise,  $V$  makes the last move, but this does not influence whether  $V$  accepts or rejects, so we could just suppress it. Let  $R$  denote  $V$ 's random tape. Denote the prover's first move by  $M$ . We call  $V$ 's response  $V(M, R)$  the "challenge" and denote it by  $C$ . We call the prover's response the "answer" and denote it by  $A$ . Then whether  $V$  accepts is a function  $V(M, C, A, R) \in \{0, 1\}$  of  $M, A, R$ .

We can also assume that  $P_x^*$  is deterministic, by viewing its random tape as fixed to one with at least half the average probability of acceptance by  $V^k$  namely  $2\epsilon/2 = \epsilon$ . (By randomly sampling an expected  $\Theta(1/\epsilon)$  random tapes for  $P_x^*$  and testing them via simulating  $V^k$  and  $P_x^*$ ,  $S$  can find a tape with at least an  $\epsilon$  conditional probability of acceptance by  $V^k$ .)

Now consider the interaction between  $V^k$  and a prover  $P^*$  for the  $k$ -fold game, on common input  $x$ . Let  $\vec{M} = (M_1, \dots, M_k)$  be  $P_x^*$ 's first move. The following algorithm THS (Trust Halving Strategy) will be used as a sub-routine by  $S$ . It takes input an index  $I \in \{1, \dots, k\}$  indicating a sub-game, a particular challenge  $C$  of the original verifier, and  $\vec{M}$ , and will either produce an answer or a special symbol  $\perp$  indicating failure.

**Algorithm THS**( $I, C, (M_1, \dots, M_k)$ )

$C_I \leftarrow C$

**For**  $j = 1, \dots, k$  **such that**  $j \neq I$  **do**

$R_j \leftarrow \{0, 1\}^r, C_j \leftarrow V(M_j, R_j)$

$(A_1, \dots, A_k) \leftarrow P_x^*(M_1, \dots, M_k, C_1, \dots, C_k)$

**For**  $j = 1, \dots, k$  **such that**  $j \neq I$  **do**

$d_j \leftarrow V(M_j, C_j, A_j, R_j)$  (a bit indicating accept or reject for the  $j$ -th sub-game)

$t \leftarrow |\{j \in [k] - \{I\} : d_j = 0\}|$

With probability  $2^{-t}$  output  $A_I$ , else output  $\perp$ .

The idea of the trust halving strategy is that  $S$  wants to use  $P_x^*$ 's answer in the  $I$ -th game as the response to challenge  $C_I$  if  $P_x^*$  is actually producing good answers. However, most of the time (namely a  $1 - \epsilon$  fraction of the time),  $P_x^*$  may produce an arbitrary mixture of answers that are and are not accepted. We use the answers on the other sub-games to decide whether or not to trust  $P_x^*$ 's answer on  $C_I$ . The trust halving strategy was introduced in the context of direct product theorems in [25].

Since THS usually produces no output,  $S$  will actually need to run it several times. Also, the prover for a single run will need to send its first message  $M_I$  before receiving a challenge, and so one good value of  $I$  needs to be found.

We usually choose  $R_I$  according to some distribution and then run  $\text{THS}(I, C, \vec{M})$  where  $C = V(M_I, R_I)$ . The THS algorithm now picks at random  $R_1, \dots, R_{I-1}, R_{I+1}, \dots, R_k$ . We refer to  $\vec{R} = (R_1, \dots, R_k)$  as the *execution base* of this run of the algorithm.

Fix  $I \in [k]$  a random tape  $R \in \{0, 1\}^r$  for  $V$ , and a sequence  $\vec{M}$  of messages from  $P_x^*$ . For  $d \in \{0, 1\}$  (that is, accept or reject) let

$$T_d(I, R, \vec{M}) = \Pr \left[ C \leftarrow V(M_I, R), A \leftarrow \text{THS}(I, C, \vec{M}) \quad A \neq \perp \text{ and } V(M_I, C, A, R) = d \right]$$

In other words,  $T_1(I, R, \vec{M})$  (resp.  $T_0(I, R, \vec{M})$ ) is the probability that THS produces an output  $A \neq \perp$ , and this answer makes the original verifier accept (resp. reject). The probability here is over the random choices of  $R_1, \dots, R_{I-1}, R_{I+1}, \dots, R_k$ , and the randomness underlying the final  $2^{-t}$  probability, made by THS. Say that  $(I, R, \vec{M})$  is *good* if

$$T_1(I, R, \vec{M}) \geq \frac{2}{\delta} T_0(I, R) + \frac{\epsilon}{16}$$



If  $(I, R, \vec{M})$  is not good, say it is *bad*. For  $I \in \{1, \dots, k\}$  let

$$\begin{aligned} \mathcal{B}(I, \vec{M}) &= \{ R \in \{0, 1\}^r \mid (I, R, \vec{M}) \text{ is bad} \} \\ b(I, \vec{M}) &= |\mathcal{B}(I, \vec{M})| \end{aligned}$$

Notice that by sampling we can estimate  $b(I, \vec{M})$  given  $I$ . (More precisely, given  $I$ , oracles for  $P_x^*$ ,  $V_x$ , and any parameters  $\epsilon', \delta' > 0$ , we can produce, in time  $\text{poly}(1/\epsilon', \log(1/\delta'), \text{Comm}((P^*)^*, V^k, x))$ , an estimate  $\tilde{b}(I, \vec{M})$  such that  $|\tilde{b}(I, \vec{M}) - b(I, \vec{M})| \leq \epsilon'$  with probability at least  $1 - \delta'$ .) For simplicity we assume henceforth that it is possible to actually compute  $b(I, \vec{M})$ . We are now ready to specify how our prover provides the answer.

**Algorithm**  $S^{P_x^*, V_x}(x, \epsilon, \delta)$

- (1)  $(M_1, \dots, M_k) \leftarrow P_x^*(x)$
- (2) By estimating  $b(I, \vec{M})$  for all  $i \in [k]$ , find a value of  $I \in [k]$  such that  $b(I, \vec{M}) \leq (\delta/4) \cdot 2^r$ . (Such a value exists by Lemma B.3.)
- (3) Send  $M_I$  to the verifier, and receive in response a challenge  $C$ .
- (4) Run  $\text{THS}(I, C, \vec{M})$  until either an output  $A$  is produced or a total of  $16 \ln(4/\delta)/\epsilon$  runs are completed. In the former case, provide  $A$  as the answer to the verifier. In the latter case, fail.

The following lemma considers the interaction between prover  $S^{P_x^*, V_x}$  and verifier  $V$  on input  $x$ , and says the probability of acceptance is quite high.

**Lemma B.1**  $\text{Acc}(S^{P_x^*, V_x}, V, x) \geq 1 - \delta$

**Proof:** From lemma B.3,  $S$  will succeed in finding an  $I \in [k]$  so that  $b(I, \vec{M}) \leq (\delta/4) \cdot 2^r$ . If  $(I, R, \vec{M})$  is good then the probability that  $S$  produces an  $A$  such that  $V(M_I CA, R) = 1$  is at least  $1 - 3\delta/4$  by Lemma B.2. Thus the overall probability that  $V$  accepts in its interaction with  $S$  is at least  $(1 - \delta/4)(1 - 3\delta/4) > 1 - \delta$ . ■

**Lemma B.2** Suppose  $S$  chooses  $I$  as the index in Step 2 and  $(I, R, \vec{M})$  is good. Let  $C = V(M_I, R)$ . Then the probability that  $S$  produces an answer  $A$  such that  $V(M_I CA, R) = 1$  is at least  $1 - 3\delta/4$ . (The probability is over the random choices of  $S$  in the steps following Step 2.)

**Proof:** Since  $(I, R, \vec{M})$  is good, we have  $T_1(I, R, \vec{M}) > (2/\delta) \cdot T_0(I, R, \vec{M}) + \epsilon/16$ . Then  $S$  makes independent simulations of  $\text{THS}$  until the latter produces some output, or until  $16(\ln(4/\delta))/\epsilon$  such simulations have been completed. The probability that  $\text{THS}$  produces some output is at least  $T_1(I, R, \vec{M}) > \epsilon/16$ . So the probability of  $S$  “timing out” without producing output, is at most  $(1 - \epsilon/16)^{16(\ln(4/\delta))/\epsilon} \leq \delta/4$ . On the other hand, since  $S$  makes independent tries of  $\text{THS}$  until output is produced, given that  $S$  produces an output, the probability that  $V$  accepts is

$$\begin{aligned} \frac{T_1(I, R, \vec{M})}{T_1(I, R, \vec{M}) + T_0(I, R, \vec{M})} &\geq \frac{T_1(I, R, \vec{M})}{T_1(I, R, \vec{M}) + (\delta/2)[T_1(I, R, \vec{M}) - \epsilon/16]} \\ &\geq \frac{T_1(I, R, \vec{M})}{T_1(I, R, \vec{M}) + (\delta/2)T_1(I, R, \vec{M})} \\ &= \frac{1}{1 + \delta/2} \\ &\geq 1 - \delta/2 \end{aligned}$$

Thus, the probability that  $S$  produces an output  $A$  and  $V(M_I CA, R) = 1$  is at least  $1 - \delta/2 - \delta/4 = 1 - 3\delta/4$ . ■

**Lemma B.3** *There is an  $I \in [k]$  such that  $b(I, \vec{M}) \leq (\delta/4) \cdot 2^r$*

**Proof:** Assume not, meaning  $b(i, \vec{M}) > (\delta/4) \cdot 2^r$  for all  $i = 1, \dots, k$ . Now for each  $i = 1, \dots, k$  fix a set  $B(i, \vec{M}) \subseteq \mathcal{B}(i, \vec{M})$  such that  $|B(i, \vec{M})|$  is *exactly*  $(\delta/4) \cdot 2^r$ . We claim that

$$\mathbf{E} \left[ i \leftarrow [k], R \leftarrow B(i, \vec{M}) \quad T_1(i, R, \vec{M}) - (2/\delta) \cdot T_0(i, R, \vec{M}) \right] > \frac{\epsilon}{16} \quad (1)$$

This implies that there exists an index  $i \in [k]$  and a string  $R \in B(i, \vec{M})$  such that  $T_1(i, R, \vec{M}) - (2/\delta) \cdot T_0(i, R, \vec{M}) > \epsilon/16$ , contradicting the fact that  $(i, R, \vec{M})$  is bad for any  $R \in B(i, \vec{M})$ , and thus proves the lemma. It remains to prove Equation (1)

We first introduce some notation. For  $\vec{R} = (R_1, \dots, R_k)$  a fixed sequence of random tapes, let

$$\begin{aligned} L(\vec{R}, \vec{M}) &= \{ i \in [k] \mid R_i \in B(i, \vec{M}) \} \\ l(\vec{R}, \vec{M}) &= |L(\vec{R}, \vec{M})| \end{aligned}$$

With  $\vec{R}$  still fixed let  $\vec{C} = (C_1, \dots, C_k)$  where  $C_j = V(M_j, R_j)$  for  $j = 1, \dots, k$ , and let  $(A_1, \dots, A_k) \leftarrow P_x^*(\vec{M}, \vec{C})$ . Now let  $T(\vec{R}, \vec{M})$  be the set of all  $i \in [k]$  such that  $V(M_i, C_i, A_i, R_i) = 0$ . Let  $t(\vec{R}, \vec{M}) = |T(\vec{R}, \vec{M})|$ .

We now define a distribution  $\mu_{\vec{M}} : \{0, 1\}^{rk} \rightarrow [0, 1]$  which assigns a probability  $\mu_{\vec{M}}(\vec{R})$  to any random tape  $\vec{R} = (R_1, \dots, R_k)$  of  $V^k$ , computed as follows

$$\begin{aligned} \mu_{\vec{M}}(\vec{R}) &= \\ \Pr \left[ i \leftarrow [k], R'_i \leftarrow B(i, \vec{M}), R'_j \leftarrow \{0, 1\}^r \text{ for } j \in [k] - \{i\} \mid (R'_1, \dots, R'_k) = (R_1, \dots, R_k) \right] \end{aligned} \quad (2)$$

This is the probability that  $\vec{R}$  is the execution base for THS in the experiment where we first pick  $I \leftarrow [k]$  and  $R_I \leftarrow B(I, \vec{M})$ , set  $C = V(M_I, R_I)$ , and then run  $\text{THS}(I, C, \vec{M})$

$$\text{Claim 1} \quad \mu_{\vec{M}}(\vec{R}) = \frac{4 \cdot l(\vec{R}, \vec{M})}{\delta k} \cdot 2^{-rk}$$

*Proof.* Refer to Equation (2) and consider the chance that the experiment in question indeed yields the particular outcome  $\vec{R}$ . For this to happen it must be that the randomly chosen  $i$  lands in  $L(\vec{R}, \vec{M})$ , and this happens with probability  $l(\vec{R}, \vec{M})/k$ . Next,  $R'_i$  must equal  $R_i$ , and this happens with probability  $1/|B(i, \vec{M})| = 2^{-r}/(\delta/4)$ . Finally it must be that  $R'_j = R_j$  for  $j \neq i$  and this happens with probability  $2^{-(k-1)r}$ . So

$$\mu_{\vec{M}}(\vec{R}) = \frac{l(\vec{R}, \vec{M})}{k} \cdot \frac{2^{-r}}{\delta/4} \cdot 2^{-(k-1)r},$$

which simplifies to the claimed quantity  $\square$

*Claim 2* Given  $\vec{R}$  drawn according to distribution  $\mu_{\vec{M}}$ , the distribution on the index  $i$  (in the experiment of Equation (2)) is uniform over  $L(\vec{R}, \vec{M})$

*Proof.* Clear  $\square$

Now we need a few more definitions. For a fixed  $\vec{R} = (R_1, \dots, R_k)$  let  $\text{THS}_{\vec{R}}(I, \vec{M})$  be the output of  $\text{THS}(I, V(M_I, R_I), \vec{M})$  when the execution base is  $\vec{R}$ . (This is a random variable depending on the random choices underlying the  $2^{-t}$  probability in the description of THS, the last being the only randomness left when the execution base is fixed.) For any  $\vec{R}$  such that  $\mu_{\vec{M}}(\vec{R}) > 0$ , and each  $d \in \{0, 1\}$ , let

$$T_d(\vec{R}, \vec{M}) = \mathbf{E} \left[ i \leftarrow L(\vec{R}, \vec{M}), A \leftarrow \text{THS}_{\vec{R}}(i, \vec{M}) \mid A \neq \perp \text{ and } V(M_i, C_A, R_i) = d \right]$$

be the chance that THS produces an output with outcome  $d$  when its execution base is fixed to  $\vec{R}$  and  $z$  is chosen at random from  $L(\vec{R}, \vec{M})$ . Let

$$X(\vec{R}, \vec{M}) = T_1(\vec{R}, \vec{M}) - \frac{2}{\delta} T_0(\vec{R}, \vec{M}) \quad (3)$$

Notice that

$$\mathbf{E} \left[ \vec{R} \leftarrow \mu_{\vec{M}} \quad X(\vec{R}, \vec{M}) \right] = \mathbf{E} \left[ z \leftarrow [k], R \leftarrow B(z, \vec{M}) \quad T_1(z, R, \vec{M}) - (2/\delta) T_0(z, R, \vec{M}) \right]$$

Thus to show Equation (1) we wish to show that

$$\mathbf{E} \left[ \vec{R} \leftarrow \mu_{\vec{M}} \quad X(\vec{R}, \vec{M}) \right] \stackrel{\text{def}}{=} \sum_{\vec{R} \in \{0,1\}^{kr}} X(\vec{R}, \vec{M}) \mu_{\vec{M}}(\vec{R}) > \frac{\epsilon}{16} \quad (4)$$

**Claim 3** Fix  $\vec{R}$  such that  $\mu_{\vec{M}}(\vec{R}) > 0$  and let  $\vec{C} = V_x^k(\vec{M}, \vec{R})$

(1) Let  $\vec{A} = P_x^*(\vec{M}\vec{C})$ . If  $V_x^k(\vec{M}\vec{C}\vec{A}, \vec{R}) = 1$  (equivalently,  $t(\vec{R}, \vec{M}) = 0$ ) then  $X(\vec{R}, \vec{M}) = 1$

(2)  $X(\vec{R}, \vec{M}) \geq \frac{2^{-t(\vec{R}, \vec{M})}}{l(\vec{R}, \vec{M})} \left[ l(\vec{R}, \vec{M}) - \left(1 + \frac{4}{\delta}\right) t(\vec{R}, \vec{M}) \right]$

(3)  $X(\vec{R}, \vec{M}) \geq -\frac{4}{\delta} 2^{-t(\vec{R}, \vec{M})}$

*Proof* For (1), note that  $V_x^k(\vec{M}\vec{C}\vec{A}, \vec{R}) = 1$  implies  $T_1(\vec{R}, \vec{M}) = 1$  and  $T_0(\vec{R}, \vec{M}) = 0$ .

Now we claim

$$T_1(\vec{R}, \vec{M}) \geq \frac{l(\vec{R}, \vec{M}) - t(\vec{R}, \vec{M})}{l(\vec{R}, \vec{M})} 2^{-t(\vec{R}, \vec{M})} \quad (5)$$

$$T_0(\vec{R}, \vec{M}) \leq \frac{t(\vec{R}, \vec{M})}{l(\vec{R}, \vec{M})} 2^{-[t(\vec{R}, \vec{M})-1]} \quad (6)$$

By Equation (3), this implies both part (2) and part (3) of Claim 3, so it suffices to show these two bounds.

To see Equation (5), consider when does  $\text{THS}_{\vec{R}}(z, \vec{M})$  produce an answer that is accepted by  $V$ , in the experiment defining  $T_1(\vec{R}, \vec{M})$ . This happens only when  $z$  lands in the set  $L(\vec{R}, \vec{M}) - T(\vec{R}, \vec{M})$ . Furthermore, since there is an acceptance in the  $z$ -th sub-game, the number  $t$  of rejections  $t$  that THS counts will equal  $t(\vec{R}, \vec{M})$ , and thus, given that  $z$  lands in the set in question, there is a  $2^{-t(\vec{R}, \vec{M})}$  chance that an accepted output is produced.

To see Equation (6), consider when does  $\text{THS}_{\vec{R}}(z, \vec{M})$  produce an answer that is rejected by  $V$ , in the experiment defining  $T_0(\vec{R}, \vec{M})$ . This happens only when  $z$  lands in the set  $L(\vec{R}, \vec{M}) \cap T(\vec{R}, \vec{M})$ , whose size is upper bounded by  $t(\vec{R}, \vec{M})$ . Furthermore, since there is a rejection in the  $z$ -th sub-game, the number  $t$  of rejections that THS counts will equal  $t(\vec{R}, \vec{M}) - 1$ , and thus, given that  $z$  lands in the set in question, there is a  $2^{-[t(\vec{R}, \vec{M})-1]}$  chance that a rejected output is produced.  $\square$

Let  $l_0 = (\delta k)/8$  and  $t_0 = l_0/(1 + 4/\delta)$ . Partition  $\{0,1\}^{kr}$  into four sets as follows

- (1) **ACCEPTED** =  $\{ \vec{R} \in \{0,1\}^{kr} \mid t(\vec{R}, \vec{M}) = 0 \}$  (In other words, those random tapes on which  $V^k$  accepts  $P_x^*$ . These give THS a non-negligible advantage.)
- (2) **EASY** =  $\{ \vec{R} \notin \text{ACCEPTED} \mid b(\vec{R}, \vec{M}) < l_0 \}$  (In other words, those random tapes without many "hard to please" components. These are rare.)
- (3) **ADVANTAGE** =  $\{ \vec{R} \in \{0,1\}^{kr} \mid b(\vec{R}, \vec{M}) \geq l_0 \text{ and } 0 < t(\vec{R}, \vec{M}) \leq t_0 \}$  (Those random tapes where  $P_x^*$  convinces the verifier on sufficiently many protocols that it is to our advantage to use one of them as our move.)

- (4)  $\text{MISTAKES} = \{ \vec{R} \in \{0,1\}^{kr} \mid b(\vec{R}, \vec{M}) \geq l_0 \text{ and } t(\vec{R}, \vec{M}) > t_0 \}$  (Those random tapes where  $P_x^*$  fails to convince the verifier on many runs of the protocol. These will almost always produce no output for THS.)

For any set  $A \subseteq \{0,1\}^{kr}$  let

$$\mathcal{S}(A) = \sum_{\vec{R} \in A} X(\vec{R}, \vec{M}) \mu_{\vec{M}}(\vec{R})$$

We analyze  $\mathcal{S}(\{0,1\}^{kr})$  by breaking it up over the four sets defined above

Suppose we flip  $k$  coins, independently, where each has probability  $p = \delta/4$  of being heads. Let  $Q$  be the probability that we get fewer than  $l_0 = (\delta k)/8$  heads. By Chernoff bounds,

$$Q < e^{-(\delta k/8)^2/(2k)} = e^{-\delta^2 k/128} \quad (7)$$

*Claim 4*

- (1)  $\mathcal{S}(\text{ACCEPTED}) \geq (\epsilon - Q)/2$
- (2)  $\mathcal{S}(\text{EASY}) \geq -Q/\delta$
- (3)  $\mathcal{S}(\text{ADVANTAGE}) \geq 0$
- (4)  $\mathcal{S}(\text{MISTAKES}) \geq -(4/\delta)2^{-l_0}$

*Proof* From the first part of Claim 3 we know that  $X(\vec{R}, \vec{M}) = 1$  for every  $\vec{R} \in \text{ACCEPTED}$ , so  $\mathcal{S}(\text{ACCEPTED}) = \Pr[\vec{R} \leftarrow \mu_{\vec{M}} \mid \vec{R} \in \text{ACCEPTED}]$  is just the probability, under distribution  $\mu_{\vec{M}}$ , of the set  $\text{ACCEPTED}$ . By assumption  $\text{Acc}(P^*, V^k, x) \geq \epsilon$  so  $|\text{ACCEPTED}| \geq \epsilon 2^{rk}$ . Let  $\text{HARDACCEPTS} = \{ \vec{R} \in \text{ACCEPTED} \mid l(\vec{R}, \vec{M}) \geq l_0 \}$ . Then  $|\text{HARDACCEPTS}| \geq (\epsilon - Q)2^{rk}$ , since  $Q$  is the fraction of sequences with  $l(\vec{R}, \vec{M}) < l_0$ . For every  $\vec{R} \in \text{HARDACCEPTS}$ , we have (using Claim 1)

$$\mu_{\vec{M}}(\vec{R}) = \frac{4l(\vec{R}, \vec{M})}{\delta k} \cdot 2^{-rk} = \frac{l(\vec{R}, \vec{M})}{2l_0} \cdot 2^{-rk} \geq (1/2)2^{-rk}$$

Thus,

$$\begin{aligned} \mathcal{S}(\text{ACCEPTED}) &= \Pr[\vec{R} \leftarrow \mu_{\vec{M}} \mid \vec{R} \in \text{ACCEPTED}] \\ &\geq \sum_{\vec{R} \in \text{HARDACCEPTS}} \mu_{\vec{M}}(\vec{R}) \\ &\geq |\text{HARDACCEPTS}| (1/2)2^{-rk} \\ &\geq (\epsilon - Q)/2 \end{aligned}$$

This proves the first part

Similarly to the first part,  $|\text{EASY}| \leq Q2^{rk}$ , and for every  $\vec{R} \in \text{EASY}$ ,  $\mu_{\vec{M}}(\vec{R}) \leq (1/2)2^{-rk}$ . Since  $X(\vec{R}, \vec{M}) \geq -2/\delta$  for any  $\vec{R}$ , we have

$$\mathcal{S}(\text{EASY}) \geq (-2/\delta) \sum_{\vec{R} \in \text{EASY}} \mu_{\vec{M}}(\vec{R}) \geq (-2/\delta) |\text{EASY}| (1/2)2^{-rk} \geq -Q/\delta$$

This proves the second part

For any  $\vec{R} \in \text{ADVANTAGE}$  we have

$$\begin{aligned} X(\vec{R}, \vec{M}) &\geq \frac{2^{-l(\vec{R}, \vec{M})}}{l(\vec{R}, \vec{M})} \left[ l(\vec{R}, \vec{M}) - \left(1 + \frac{4}{\delta}\right) t(\vec{R}, \vec{M}) \right] \quad (\text{by Claim 3}) \\ &\geq \frac{2^{-l(\vec{R}, \vec{M})}}{l(\vec{R}, \vec{M})} \left[ l_0 - \left(1 + \frac{4}{\delta}\right) t_0 \right] \quad (\text{because } t(\vec{R}, \vec{M}) \leq t_0 \text{ and } l(\vec{R}, \vec{M}) \geq l_0) \\ &= 0 \quad (\text{because } t_0 = l_0/(1 + 4/\delta)) \end{aligned}$$

This proves the third part

From part three of Claim 3, for any  $\vec{R} \in \text{MISTAKES}$  we have  $X(\vec{R}, \vec{M}) \geq -(4/\delta)2^{-l(\vec{R}, \vec{M})} \geq -(4/\delta)2^{-l_0}$   
This proves the fourth part

This concludes the proof of Claim 4  $\square$

Now we can apply Claim 4 to get

$$\begin{aligned}
\mathcal{S}(\{0,1\}^{rk}) &= \mathcal{S}(\text{ACCEPTED}) + \mathcal{S}(\text{EASY}) + \mathcal{S}(\text{ADVANTAGE}) + \mathcal{S}(\text{MISTAKES}) \\
&\geq (\epsilon - Q)/2 - Q/\delta - 0 - (4/\delta)2^{-l_0} \\
&\geq \epsilon/2 - Q/2 - Q/\delta - (4/\delta)e^{-\frac{\delta k/8}{1+4/\delta}} \\
&= \epsilon/2 - Q/2 - Q/\delta - (4/\delta)e^{-\frac{\delta^2 k}{8(4+\delta)}} \\
&\geq \epsilon/2 - Q/2 - Q/\delta - (4/\delta)e^{-\frac{\delta^2 k}{40}} \quad (\text{because } \delta < 1) \\
&> \epsilon/2 - Q/2 - Q/\delta - (4/\delta)e^{-\frac{\delta^2 k}{128}}
\end{aligned}$$

The assumption in the statement of Theorem 4.1 is that  $\epsilon > (16/\delta)e^{-\delta^2 k/128}$ . Combining this with Equation (7) yields  $Q < e^{-\delta^2 k/128} < \delta\epsilon/16 < \epsilon/16$ . Using these estimates in the above we get

$$\begin{aligned}
\mathcal{S}(\{0,1\}^{rk}) &\geq \epsilon/2 - \epsilon/32 - \epsilon/16 - (4/\delta)(\delta\epsilon/16) \\
&> \epsilon/2 - 6\epsilon/16 \\
&> \epsilon/8
\end{aligned}$$

This yields Equation (4), which we had already said yields Equation (1), and thus concludes the proof of Lemma B.3  $\blacksquare$

## C Parallel repetition when the verifier has secret information

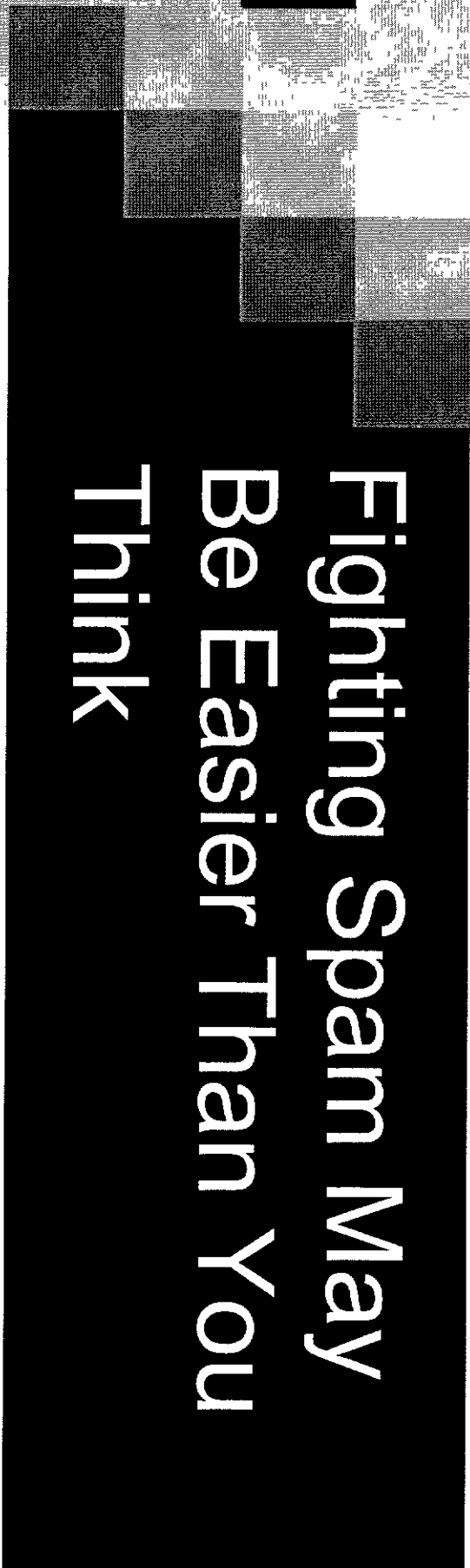
In our model the verifier is not given any secret information pertaining to the input  $x$  or to the context  $\lambda$ . Its strategy must be a computable (in probabilistic) polynomial time given  $x, \lambda$ . This is important to make the parallel repetition question meaningful as we now explain.

If we were to allow the verifier access to private information about  $x$  or  $\lambda$  then it is trivial to see that parallel repetition fails to lower the error. In fact one can easily specify a two round protocol which has error  $1/2$  but, when repeated  $k$  times in parallel, the error actually increases with  $k$ , tending to 1. For example, say  $x$  (or  $\lambda$ ) is a random integer product of two primes  $p, q$  such that  $V$  knows  $p, q$ . Let the protocol be that  $V$  flips a coin and sends in its first message  $p, q$  if the coin was 0 and nothing if the coin was 1. It accepts if the prover's reply is  $p, q$ . The error probability of this protocol is  $1/2 + \nu(\cdot)$  where  $\nu(\cdot)$  is negligible. However, if we repeat the protocol  $k$  times there is a strategy for the prover to win with probability  $1 - 2^{-k}$ , because except with probability  $2^{-k}$  the factorization  $p, q$  is released by the verifier in *some* run, and then the prover can echo it in all runs.

Is it reasonable that we deny the verifier secret information? Certainly it seems reasonable that the verifier has no secret information about  $x$ , because in cryptographic settings  $x$  is chosen by the (honest) prover and given to  $V$ , and the prover is trying to prove something about  $x$  to  $V$ , so the latter would not have any information about  $x$  other than what he could compute in polynomial time given  $x$ , which is what we assume. It is possible that  $V$  does have secret information related to  $\lambda$ . (For example the latter could contain his public key, of which he holds the corresponding secret key.) Still, it makes sense that the protocol, whose goal is to prove a claim about  $x$ , does not use this

secret information, so that as far as the protocol is concerned, we can assume the verifier strategy does not depend on that secret information. Certainly all “useful” protocols fall in our model, we are eliminating only artificial ones.





Fighting Spam May  
Be Easier Than You  
Think

Cynthia Dwork  
Microsoft Research SVC



# Why?

## ■ Huge problem

Industry: costs in worker attention, infrastructure

Individuals: increased ISP fees

Hotmail: huge storage costs, 65-85%

FTC: fraud, confidence crimes

Ruining e-mail, devaluing the Internet

## Computational Approach [DN'92]

- If I don't know you:  
Prove you spent ten seconds CPU time,  
just for me, and just for this message
- User Experience:  
Automatically and in the background  
Checking proof extremely easy
- All unsolicited mail treated equally

# Principal Techniques

## ■ Filtering

- Everyone: text-based
- Brightmail: decoys; rules updates
- Microsoft Research: (seeded) trainable filters
- SpamCop, Osirusoft, etc: IP addresses, proxies, ...

## ■ Make Sender Pay

- Computation [Dwork-Naor'92; Back'97]
- Human Attention [Naor'96, DEC patent]
- Money [eg, Gates'96, Hayes, McCurley]

# Outline

- The Computational Approach
  - Overview; economic considerations
  - Burrows' suggestion: memory-bound functions
- Turing Tests
- Architectures
  - Point-to-Point
  - Web-based mail
- Deeper Discussion of Memory-Bound Functions
- Nascent Proposal (joint with A. Goldberg and M. Naor)

# Economics

- $(80,000 \text{ s/day}) / (10\text{s/message}) = 8,000 \text{ msgs/day}$
- Hotmail's billion daily spams:
  - 125,000 CPUs
  - Up front capital cost just for HM: circa \$150,000,000
- The spammers can't afford it.

# Who Are the Spammers?

"Most of the spammers are not wealthy people," said Stephen Kline, a lawyer for the New York State attorney general's office.

# Who Are the Spammers?

- Spamhaus ROKSO 100/90%
- F. Krueger, SMN: circa 300 people total; very top few spammers make a few million/year; resources spent “dealing with” ISPs, setting up shell companies
- Big Player: eUniverse (Yahoo! lists annual earnings of \$6.1 Mil), has mailing list of 100,000,000 names, Mosaic Data Systems has list of 60,000,000

# Cryptographic Puzzles

- Hard to compute;  $f(m, S, R, t)$  can't be amortized  
lots of work for the sender
- Easy to check " $z = f(m, S, R, t)$ "  
little work for receiver
- Parameterized to scale with Moore's Law  
easy to exponentially increase computational cost, while  
barely increasing checking cost
- Can be based on (carefully) weakened signature  
schemes, hash collisions



# Burrows' Suggestion

- CPU speeds vary widely across machines, but memory latencies vary much less (10+ vs 4)
- So: design a puzzle leading to a large number of cache misses

# Social Issues

- Who chooses  $f$ ?  
One global  $f$ ? Who sets the price?  
Autonomously chosen  $f$ 's?
- How is  $f$  distributed (ultimately)?  
Global  $f$  built into all mail clients? (1-pass)  
Directory? Query-Response? (3-pass)

## Computation: Technical Issues

- Distribution Lists (!)
- Awkward Introductory Period  
Old versions of mail programs; bounces
- Very Slow/Small-Memory Machines
- Cache Thrashing (memory-bound)
- The Subverters

# Turing Tests: Payment in Kind [N'96]

- CAPTCHAs (Completely Automated Public T. test for telling Computers and Humans Apart)
  - Defeat automated account generation
  - 5-10% drop in subscription rate
  - Teams of conjectured-humans (8-hour shifts)
- Yes: Distorted images of simple word
- No: ``Find 3 words in image with multiple overlapping images of words"
- Others: subject classification, audio
- M. Blum: people have done preprocessing

# Social and Technical Issues

- Social (especially in enterprise setting)
  - ADA, S.508 (blind, dyslexic)
  - Not ``professional''
  - Productivity cost: context switch (may be mitigated in architecture supporting pre-computation)
  - Wrong direction: we should offload crap onto machines
- Technical
  - No theory; if broken, can't scale.
  - Idrive, AltaVista, broken [J]

## Turing Test Economic Issues

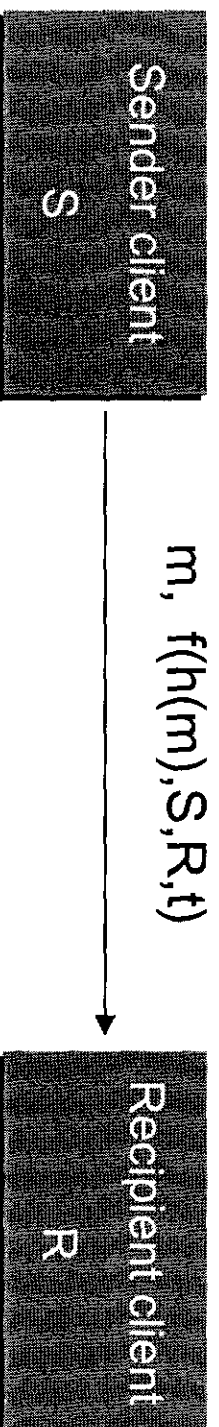
- Human time in poor country has roughly same cost as computer time (even if 8/5 wk)
  - Also need computer, but it can be slow, cheap
  - 10s same ballpark as some Turing tests
- Suppose we're wrong about cost, and really the price should be 1 cent
  - 1-cent challenge costs 2 minutes. No one *not* being paid would dream of submitting to this.

# Cycle Stealing

- Stealing cycles from humans: Pornography companies require random users to solve a CAPTCHA before seeing the next image [VABL]
- Worse for computational challenges
- Economics: lots of currency means currency is devalued. Prices go up.
- There are lots of cycles, but anyone can buy them. Can go into business brokering cycles.



# Point-to-Point Architecture



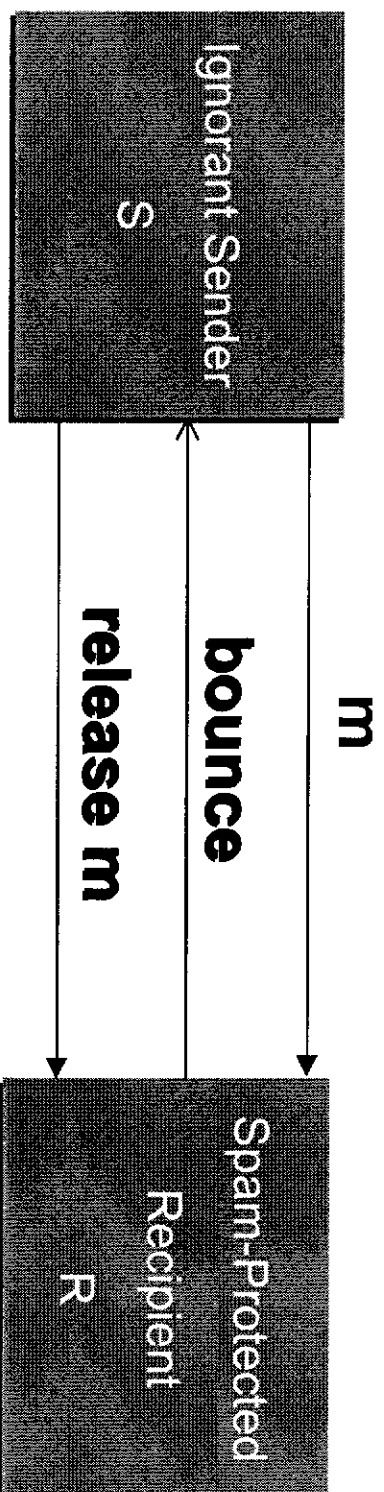
## (Ideal Message Flow)

Single-pass "send-and-forget"

Can augment with amanuensis to handle slow machines

Can add post office / pricing authority to handle money payments

# Here to There (and There ' )



- Three e-mail messages
- R's mail client caches  $m$ ,  $h(m)$ ,  $S$ ,  $R$ ,  $t$
- Bounce (viral marketing opportunity)
  - html attachment with Java Script for  $f$  (make it an intrinsic)
  - contains parameters for  $f$  ( $h(m)$ ,  $S$ ,  $R$ ,  $t$ )
  - clicking on link causes computation, sending e-mail (optional) link for download of client software

## **Remark: Web-Based Mail**

- Computation done by client (applet or JavaScript)
- Verification and safelist checking done by server

# Memory-Bound Functions [ABMW'02]

## ■ Devilish model!

- assume cache size (of spammer's big machine) is half the size of memory (of weakest sending machine)
- must avoid exploitation of locality, cache lines (factors of 64 or even 16 matter)
- watch out for low-space attacks in crypto literature

# Incompressibility and RC4 [DGN'02]

- Big ( $2^{22}$ ) Fixed-Forever Truly Random T
- Model walk on prg of RC4 stream cipher
- Intuition:

Use the pr bit stream (seeded by message) to choose entries of T to access.

These, in turn, affect the stream (defending against pre-processing to order T so as to exploit locality).

# RC4 (Partial Description)

Initialize  $A$  to pseudo-random permutation of  $\{1, 2, \dots, 256\}$  using a seed  $K$

■ Initialize Indices:

$i = 0; j = 0$

■ PR generation loop:

$i = i + 1$

$j = j + A[i]$

Swap ( $A[i], A[j]$ );

Output  $z = A[A[i] + A[j]]$

# DGN: Initializing $A$ and Basic Step

- Fixed-Forever truly random  $A$  of size 256; 32-bit words
- Initialize Indices:  
 $i=0; j=0$
- Walk:
  - $i = i + 1$
  - $j = j + A[i]$
  - $A[j] = A[j] \odot T[c]$
  - Swap ( $A[i], A[j]$ )
  - $c = T[c] \odot A[A[i] + A[j]]$
- Given  $m$  and trial number  $i$ , compute strong cryptographic 256-word mask  $M$ .
- Set  $A = M \odot A$ .
- $c = A \bmod 2^{22}$



# Side by Side

Generate pseudo-random permutation of  $\{1, 2, \dots, 256\}$  using a seed  $K$

- Fixed-Forever truly random  $A$  of size 256; 32-bit words
- Given  $m$  and trial number  $i$ , compute strong cryptographic 256-word mask  $M$ .
- Set  $A = M \odot A$ .
- $C = A \bmod 2^{22}$

# Side by Side

- Initialize Indices:

$i = 0; j = 0$

- PR generation loop:

$i = i + 1$

$j = j + A[i]$

- Initialize Indices:

$i=0; j=0$

- Walk:

$i = i + 1$

$j = j + A[i]$

$A[i] = A[i] \odot T[c]$

Swap ( $A[i]$ ,  $A[j]$ )

$c = T[c] \odot A[A[i] + A[j]]$

Swap ( $A[i]$ ,  $A[j]$ )

Output  $z = A[A[i] + A[j]]$

# Full DGN

- E: factor by which computation cost exceeds verification
- L: length of walk
- Trial  $i$  succeeds if hash of  $A$  after  $i$ th walk ends in  $1 + \log_2 E$  zeroes (expected number of trials is  $E$ ).
- Verification: trace the walk ( $L$  memory accesses)

# Parameters

- Want  $ELt=P$ , where

$L$  = path length

$P$  = target real time cost of proof of effort = 10s

$t$  = memory latency = 0.2 nanoseconds

- Choose  $E < |Cache| / |Cache Line|$

- Reasonable choices:

$E = 32,000, L = 625$

# Summary

- Discussed computational approach, Turing tests, economics, cycle-stealing
- Briefly mentioned two architectures
- Examined difficulties of constructing memory-bound pricing functions and proposed a new one designed to avoid these difficulties (no proofs yet)

# Future Work and Open Questions

- Implement and test Outlook, Pine plug-ins (at Stanford); add signatures
- Further study of DGN algorithm
- Distribution Lists
- Good MB functions with short descriptions (will subset product work)?







## Programmer writes spam bomb

By Janet Kornblum  
Staff Writer, CNET News.com  
August 6, 1997, 7:00 PM PT

Ron Guilmette is a self-appointed soldier in the battle against spam. Now he thinks he's developed the ultimate weapon.

"It's a war," said Guilmette, a software engineer who has been working on an antispam program DeadBolt through E-Scrub Technologies. "What I've been trying to do is devise the equivalent of a nuclear weapon for our side and we've just done it."

E-Scrub works by filtering out domain names of known spammers and constantly updating the lists dynamically to the computers of whatever individuals or Internet service providers sign up for it.

Of course, Guilmette is not the first person to come up with an antispam program. In fact, this program is very similar to America Online's PreferredMail filter on its proprietary service. But there aren't many out there who have made the development of such software so personal. He just might be one of most dedicated Netizens when it comes to getting rid of junk email.

How dedicated? Just do a search on DejaNews and you'll see just how obsessed. Since June 18, he's posted 798 messages to "news.admin.net-abuse.email," 531 to "news.admin.net-abuse.misc," and 83 to "comp.mail.misc." The subject on most of those postings? Spam.

Guilmette and a cadre of other spam haters have taken it upon themselves to hunt down spammers and do the virtual equivalent of running them out of town. They see a piece of junk email, trace it back to its source, then tell the Internet service provider that one of its members is violating antispam rules.

"I did it for a long time," he said. "I did it for months and months. I used to call ISPs, hassle them, and get them to nuke the accounts of spammers. We have the power but it takes a lot of time to track down the right ISP, to get through to the right person on the phone. It's expensive in time and long distance phone charges. You have to keep doing it for each spammer. It's too labor intensive."

However, the next best thing, he added, is his program. Of course.

advertisement

Get  
**\$100 off**  
a Gateway®  
500s Desktop

now only  
**\$899**  
(limited time offer)  
[get the details](#)



SHIPPING ALWAYS INCLUDED  
on all home systems

### Search News

Advanced search

### Latest Headl

display on desktop

Net heavyweights

Research firm calls  
innovation

Scientists protest E  
patents

Technology spendi  
growth

Chips sales up a h

Price cuts nip at C

Server market sho

Corporations seek  
results

IBM stretches gr

Dell's new PC to ar

Week ahead Netw

Webby Awards cer

Sun touts technical

Desktop cords coul  
to moon

Monster backpedal  
purge

Samsung investing  
plant

Red Hat warns to  
plan

Are file traders nex

Opening Windows

New Windows stirs

This week's headl

### News Tools

Get news by m

**XML** What is

Content licensi

Display news c

[Print story](#)

[E-mail story](#)

[Order reprints](#)

[Send us news tips](#)

### Related News

- Antispam plan discounted April 25, 1997

CNET News.com  
News

☒ Enterprise  
Senior edit  
Kanellos c

- [ISP: Internet spam provider](#) February 18, 1997
- [ISP wins round against junk mailer](#) October 24, 1996
- [Get this story's "Big Picture"](#)

#### Also from CNET Networks

- [Access thousands of high-paying IT jobs now](#)
- [Products you want from places you trust](#)
- [Essential upgrades for Windows XP](#)
- [Learn TCP/IP from professionals](#)
- [Learn how to troubleshoot your PC](#)

servers, ar  
hardware t  
business i

☒ **Daily Disp**  
Our award  
editors del  
stones rgt  
inbox (dar

Your e-mai

[Sign me up!](#)

All News.com nev

[Send us news tips](#) | [Contact Us](#) | [Corrections](#) | [Privacy Policy](#)



**Featured services** [Job Search](#) | [IT Community](#) | [Compare Prices](#) | [IT/IS Policies](#) | [MS Access](#)

CNET Networks [Builder.com](#) | [CNET](#) | [GameSpot](#) | [mySimon](#) | [TechRepublic](#) | [ZDNet](#)

Copyright ©1995-2003 CNET Networks, Inc. All rights reserved





CNET tech sites Price comparisons | Product reviews | Tech news | Do

Front Page

Enterprise

E-Business

Communications

Personal Technology

## ISP: Internet spam provider

By CNET News.com Staff  
February 18, 1997, 5 15 PM PT

You might call it spam heaven

Cyber Promotions, tired of getting the boot from unhappy Internet service providers for sending out mass email, has found a perfect solution to the spammer's dilemma. On March 17, it will open its own ISP where junk emailers will not only be accepted, but welcomed with open arms.

Sanford Wallace, president of Cyber Promotions, calls it a "bullet proof dial-up network" where, for \$50 a month, customers can send mass unsolicited commercial email with impunity "as long as they honor 'remove' lists and adhere to all local, state and federal laws," according to Wallace.

Cyber Promotions has been kicked off of at least 20 ISPs in less than three years. But Wallace said he solved that ISP problem several months ago by getting bandwidth from several different small providers.

But Ron Guilmette, a software engineer developing an antispam program called DeadBolt, which can be used to block spam-generating domain names, uses decidedly less flattering terms to describe Wallace's venture.

Guilmette, who has made stopping spam on the Net his life's work (his passion dates to 1995 when he received a neo-Nazi spam), said Wallace will continue to have a hard time finding the bandwidth to launch a service.

"He can set up some little podunk attempt, but soon as he does, his upstreams are going to get the message loud and clear that you just signed up a loser," Guilmette said. "He'll be on for a little while. Ultimately, you might as well tell him he just might give it up now because he's never going to find a permanent home. The reality is he's been chased out of everywhere he's hooked up to. We'll continue to chase him until there are no dark corners for him to hide."

Guilmette added that he intends to use his program, to be released in mid-April, to filter out any addresses coming from Wallace's new ISP. It will be free for noncommercial individual use and to Linux users.

Like America Online's (AOL) antispam program, PreferredMail, DeadBolt can block by domain name. But, Guilmette said, it can also block by IP address. And IP addresses take more time to get than domain names.

Wallace said he has plenty of IP addresses. But, he hastened to add, that didn't matter because Cyber Promotions has no desire to send email to people who don't want it.

"If it's the end user blocking, we have no problem with that," Wallace said. He objects to systems that

### Search News

Advanced search

### Latest Headl

display on desktop

Research firm calls  
[innovation](#)

Scientists protest E  
patents

Technology spendi  
growth

[Chips sales up a h](#)

[Price cuts nip at Ce](#)

[Net heavyweights i](#)

[Server market sho](#)

[Corporations seek  
results](#)

[IBM stretches grid](#)

[Dell's new PC to ap](#)

[Week ahead Netw](#)

[Webby Awards cer](#)

[Sun touts technical](#)

[Desktop cords coul  
to moon](#)

[Monster backpedal  
purge](#)

[Samsung investing  
plant](#)

[Red Hat warms to  
plan](#)

[Are file traders nex](#)

[Opening Windows](#)

[New Windows stirs](#)

[This week's headl](#)

### News Tools

Get news by m

[XML](#) What is

Content licensi

Display news c

CNET News.co  
News

☒ Enterprise  
Senior edit  
Kanellos c

block email as a whole, without letting the individual decide.

If a user wants to block mail, Wallace said, that's fine with him "We only want mail to go to people who want to get it. Believe it or not, we've never wanted to send mail to people who don't want to get it."

He added that he wouldn't exactly call his service a spammer's paradise "I'd just say it's a free world."

To people like Guilmette who say that Cyber Promotions' day has come and gone, Wallace added "Some people said the same thing about Madonna. She's [still] a very big star."

servers, at  
hardware t  
business i

☒ **Daily Disp**  
Our award  
editors del  
stones ngt  
inbox (dal

Your e-mal

**Sign me up!**

All News.com nev

[Print story](#)

[E-mail story](#)

[Order reprints](#)

[Send us news tips](#)

#### **Related News**

- [Another setback for spam king](#) February 3, 1997
- [Nevadans against spam](#) January 20, 1997
- [Spam king strikes out in court](#) December 13, 1996
- [Spam king challenged](#) December 5, 1996
- [Email spammer won't quit](#) November 7, 1996
- [Junk emailer down for the count](#) November 4, 1996
- [Mass emailer back on Sprint temporarily](#) October 30, 1996
- [Get this story's "Big Picture"](#)

#### **Also from CNET Networks**

- [Access thousands of high-paying IT jobs now](#)
- [Products you want from places you trust](#)
- [Essential upgrades for Windows XP](#)
- [Learn TCP/IP from professionals](#)
- [Learn how to troubleshoot your PC](#)

[Send us news tips](#) | [Contact Us](#) | [Corrections](#) | [Privacy Policy](#)



**Featured services:** [Job Search](#) | [IT Community](#) | [Compare Prices](#) | [IT/IS Policies](#) | [MS Access](#)

CNET Networks [Builder.com](#) | [CNET](#) | [GameSpot](#) | [mySimon](#) | [TechRepublic](#) | [ZDNet](#)

Copyright ©1995-2003 CNET Networks, Inc. All rights reserved.





[Advanced Groups Search](#) [Preferences](#) [Groups Help](#)

The Deadbolt tm +spam

[Google Search](#)

12 May 1981 - 10 Aug 1997

"The" is a very common word and was not included in your search [\[details\]](#)

## Groups search result 2 for The Deadbolt tm +spam

**Anti-Spam for Business** • Choose from leading hardware & software email filtering solutions • [www.nwtechusa.com](#)

Sponsored

**Eliminate Annoying Spam** • Want to kill time-wasting junkmail? Get iHateSpam PC World's "Best Buy" • [www.sunbelt-software.com](#)

[Links](#)

**DynaComm!mail** • Spam filtering from FutureSoft. Download your trial today • [www.dcseries.com](#)

From: [Ronald F. Guilmette \(rfg@monkeys.com\)](#)

Search Result 2

Subject: Re: To Mung or Not to Mung

Newsgroups: [news.software.readers](#), [news.admin.net-abuse.email](#)

[View Complete Thread \(48 articles\)](#)

Date: 1997/07/24

[Original Format](#)

In article <33d5b7fd.59311486@news.primenet.com>,  
Guy Tyler <guyt@primenet.com> wrote  
>>On Wed, 16 Jul 1997, Charles Arthur wrote.  
>>> Think harder. I did, and had an epiphany. Spammers almost always send on  
>>> Bcc lists. \*\*\*Your address is not in the To or Cc. headers.\*\*\*  
>>> That's very easy to filter on.  
>>> So, set up your mail program to allow all your regular Bcc mail (mailing  
>>> lists etc) and then chuck out anything which is Bcc:'d or has  
>>> "Apparently-To" in the header. They are always **spam**  
>  
>I am on an important mailing list. There are 16 of us who regularly  
>receive email from the administrator of sorts. When he sends out the  
>mail, the 16 addresses are \*always\* in "Apparently-To" headers. I would  
>like to set up filters for my mail program, Eudora Pro, but I'm afraid  
>if I filter out all mail that does not contain my address in the "To"  
>or "Cc" fields, I will miss this very important email. I also don't  
>feel I should ask the administrator to change the manner in which he  
>emails us. I don't want to inconvenience him.

The **Deadbolt(tm)** Personal E-mail Filter has a very simple solution to this problem.

It will allow you to turn on various "generic filters", like the one for blind carbons. Each time a message runs afoul of one of the generic filters that you have enabled for your personal account, the message becomes "distrusted" (as opposed to "blacklisted"). Distrusted messages are normally bounced back to the sender with a polite note telling the sender that he/she/it should send the message again, only this time including your personal pass phrase in the Subject: line. If the sender does that, then the message will get through and also, the sender's address will be automatically incorporated onto your personal E-mail Address Whitelist so that the sender will never again have to go through this registration protocol in order to reach you in the future.

In **Deadbolt**, your personal whitelists (i.e. your E-mail Address Whitelist, your Domain Name Whitelist, your IP Address Whitelist, and your Mail Header Whitelist) all take precedence over any generic filters you have enabled for your account. So using **Deadbolt**, you could go ahead and enable the Blind Carbon Copy Generic Filter (which, all by itself, would relieve you of most junk E-mail) but at the same time you could add the origination addresses for any mailing lists that want to receive traffic from to your personal E-mail Address Whitelist. Doing so would insure that you would still receive all of the mailing list traffic for the specific mailing lists that you consciously have decided to receive from, even though generally speaking, you are still not accepting Blind Carbons sent from parties not already known to you who haven't (or who don't) go through



the simple registration protocol using your personal pass phrase.

This is the best of both worlds . you get generic filtering of Blind Carbons, but you also have the ability to specify your own specific exceptions to the general filtering rules via your personal whitelists

Oh yes. . and as you might have gathered, **Deadbolt** also provides a complete set of blacklisting capabilities. You can blacklist by full E-mail address, by whole domain names, by IP address, by IP address ranges, or by specific E-mail headers and their contents **Deadbolt's** complete parsing of all important E-mail headers (most notably the Received: headers) allows for fully effective blacklisting of known rogue IP addresses even in cases where the spammers have done everything imaginable to mask or hide the true origin of their spew. (Many spammers change domain names faster than they change their shorts, but it is far more difficult, costly, and time consuming for them to change their IP addresses, so IP address are the best basis on which to lockout known stable point sources of E-mail **spam** Dis-trusting blind carbons gets most of the rest, e.g. most of the spew from the hit-and-run artists using throw-away accounts.)

--

-- Ron Guilmotte, Roseville, CA ----- E-Scrub Technologies, Inc. -----  
---- E-mail: rfg@e-scrub.com ----- <http://www.e-scrub.com/> -----  
----- Copyright (c) 1997 by Ronald F. Guilmotte, All rights reserved -----

[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

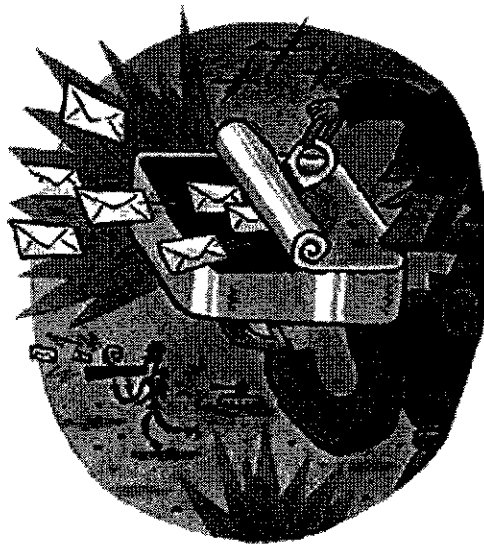
©2003 Google



s a l o n \_ m a g a z i n e + a b o u t \_ 2 1 s t + n e w s l e t t e r +



## s p a m b o m b e r s



Tired of receiving dozens of get-rich-quick offers and promos for "bulk mailers" in your e-mail? Meet the software designers who have made it all possible

BY ANDREW LEONARD | I figured Neil Albala as a hard man to get a hold of I knew he was the author of Floodgate and Goldrush, two of the Internet's first commercially available "bulk e-mail" software programs. But I also knew that the very words "bulk e-mail" are an invitation to a flame war

Bulk e-mail is a euphemism for "spam" -- that all-too-familiar influx of junk e-mail that clogs electronic mailboxes the way kudzu strangles a forest. On the Net, anti-spam vigilantes tend not to look kindly upon programmers who write software that facilitates the transport of "unsolicited commercial e-mail." Incidents of harassment -- phone calls, lawsuit threats -- are common. Albala, I had been told, practically "invented the bulk e-mail business", my guess

was that he would be lying low

I guessed wrong To my surprise, Albala answered my first e-mail query promptly -- and then proceeded to upset my expectations once again I had assumed that he would be gratified at how healthy the bulk e-mail business seems right now, if my own stuffed e-mail box is any evidence But when I asked Albala how he saw the future, he sounded downright glum

Bulk e-mail, says Albala, is under siege A flurry of spam regulation bills looms on both federal and state legislative levels A string of recent court decisions is steadily eroding spammer freedom Blocking software installed at large Internet service providers like America Online and CompuServe is "making it almost impossible for people to get the mail out " And worst of all, this spring, sales of Floodgate, which retails for about \$400, suddenly plummeted

"It's the end of an era," says Albala

Most spam-ridden users today might snort at that idea Spam stats are hard to come by, but a wealth of anecdotal evidence suggests that bulk e-mail is booming A steady stream of complaints dot Usenet newsgroups, mailing lists and online conferencing systems A systems administrator at one Internet service provider reports that 25 percent of his own mail is spam -- and the numbers keep rising Indeed, over the course of reporting this story, my daily spam count nearly doubled, as if just the act of thinking about spam exerted a magnetic attraction on "Lotto Buck\$ 4 U!!!" e-mail messages

Albala's troubled bottom line has another, more likely explanation competition Floodgate may have been the first spam software product, but over the last year, a slew of increasingly powerful "address extractor" and "bulk mailer" programs have joined it in the market, flaunting gaudy, inflammatory names like Extractor Pro, Stealth Mass Mailer, CyberBomber, E-Mail Blaster and Web Weasel

And those are just the most public weapons in the spam arsenal A thriving not-so-underground spam economy flourishes in the interstices of the Net There is a steady demand for "cloaking software" -- programs that disguise, or hide, spammer identity -- as well as for arcane little programs like "mailing list cleaners" that cull "bad" addresses from mailing lists And finally, of course, there are the address lists themselves -- for prices ranging from \$19 95

to several hundred dollars, you too can purchase 25 million e-mail addresses

Indeed, as I looked more closely at my daily dose of spam, I noticed that a significant portion, if not the majority, of my junk e-mail advertised tools designed to create more junk e-mail Welcome to the ultimate post-industrial "service economy"! On the Net, everyone is in business selling business supplies And the trickle-down effect of a vigorous spam software marketplace is, inevitably, more spam

One can hope that before we're completely overwhelmed, we'll see the development of equally powerful anti-spam tools But right now, spamming technology has the upper hand, and it is feeding on itself in a mad, spiraling frenzy

And the spammers rejoice Contrary to my initial assumption, I found that as a class, the authors of spamming tools and marketers of spamming services are a brash and highly visible crew They stand united by a firm belief that any publicity is good publicity, by an absolute refusal to acknowledge that the tools that they are creating are being used irresponsibly, and by a strong wish that the government keep its hands off their business As one reseller of the address extractor program NetContact told me, "Everything we've come to enjoy and take for granted in the Western world is at risk if we restrict the honest expression of salespeople "

Neil Albala's competitors airily dismiss his fears Adverse legislation? No problem, we'll just move our servers to another country As Forrest Dayton, author of the Stealth Mass Mailer program observes, "They couldn't ban porno on the Net How are they going to ban bulk e-mail?"

Blocking software? So what -- it's just another market opportunity Floodgate's woes, say competitors, are due to its own deficiencies Newer, smarter programs have become the spammer's first choice "My mail gets through," asserts Dayton

Depressed sales? Business has never been better, claims self-proclaimed "spam king" CyberPromo's Sanford Wallace

"Since the day we started this business we haven't shown a loss," says Wallace, who enjoys referring to himself by the nickname "Spamford " Claiming revenues this year exceeding \$2 million, he scoffed at assertions that the bulk e-mail biz is little more than a pyramid scheme in which the irresponsible rip off the clueless "You'd have to make a

mistake not to make money in this business "

CyberPromo, boasts Wallace, is responsible for "80 percent of the bulk e-mail currently delivered across the Internet " Wallace is eager to claim that CyberPromo moves 25 million e-mail messages a day for 11,000 clients But CyberPromo markets no actual products itself -- just the tools of the spamming trade The company is a one-stop-buys-all spam clearinghouse, addressing, says Wallace, the biggest problem that would-be spammers face -- the likelihood that they will lose their account at their local Internet service provider if they abuse spam policies

That's not a problem at CyberPromo, which -- in addition to its flagship product, the bulk e-mailer Cyber-Bomber -- also has its own network of mail server computers, and purchases connectivity to the Internet through a service provider, Agis, that remains deaf to anti-spammer outrage

Wallace's competitors aren't crazy about his devil-may-care stance To them, he is undermining the "respectability" of the bulk e-mail business

"His 'how are you going to stop me' attitude is great for his publicity and great for his marketing, but as far as legitimizing this business and this industry, it's not helping a whole lot," says Todd Farmer, marketing director at Extractor Marketing, which sells the address extractor programs Extractor Pro and Web Weasel

But is the business really legitimate in the first place? To most systems administrators and anti-spammers, the bulk e-mailers are taking advantage of technical loopholes in the Internet infrastructure to get a free marketing ride on an already overburdened system And the fact that they seem mostly to be selling their own tools exposes their hollowness, argues Scott Mueller, a systems administrator at WENET, a mid-sized Internet service provider in San Francisco.

"If you think about it, if there were money to be made spamming ads, then wouldn't you want to keep other people from competing with you?" asks Mueller.

Mueller is a leading force in the Coalition Against Unsolicited Commercial e-mail (CAUCE), a group that is backing federal legislation that seeks to financially penalize advertisers for unsolicited e-mail, rather than go after vendors of spamming services He's given up on techno-fixes to the spam problem, convinced that there is no way, "unless we have true artificial intelligence," that spam blocking

software can ever keep up

First, there is the endless creativity of the spammers. Spammers change their return addresses constantly, and are always coming up with new formulations of the basic pitch. And software like the Stealth Mass Mailer makes it possible to use any mail server computer on the Net as a bulk e-mail relay station, even if the bulk e-mailer doesn't have an account on the service provider that owns the mailer. Such a practice, referred to as "hijacking," is considered one of the more egregious spamming sins. Even Dayton, the author of the program, says he frowns upon such "unethical" activity.

"I don't police what my people use it for," says Dayton. "I tell them up front you shouldn't mail through a server you don't have an account from. You can open up an account with a bulk e-mail Internet service provider and you're doing it totally legit. All the program was designed for was speed."

To anti-spammers, such assertions reek of disingenuousness. But questions of morality aside, the truth is that the open architecture of the Internet permits all kinds of abuses. Decentralized anarchy is one of the Net's greatest strengths; but it doesn't come for free. If you want the power of e-mail - cheap, fast and universal -- you must pay the price of spam.

Or at least that's how it seems right now, with the prospect of any significant legislation taking effect still at least a year away, according to Mueller. But the obvious health of the market for spam tools begs an obvious question -- where is the market for spam blockers?

In its infancy, argues software engineer Ron Guilmette

Guilmette is a leading anti-spam radical -- a longtime spam hater who finally decided to bet his talent on an emerging market for anti-spam technology. Guilmette notes that new versions of popular e-mail programs such as Pegasus and Eudora come equipped with powerful and increasingly easy-to-use filter systems that allow users to configure their applications to reject unwanted mail. The problem, he concedes, is that spammers move too fast for users to keep up with -- you'd have to spend more time updating your filters than it would take just to delete the messages by hand.

Guilmette believes he has an answer to the problem -- a spam blocking program that is configurable by users, but constantly updates its own internal list of spam offenders by connecting to a central database. Guilmette says his program, Deadbolt, available for purchase in September, would be

installed at the mail server level -- thus catching, and blocking, spam before it ever is downloaded into a home computer

And just as the spammers are constantly harvesting the Net for new e-mail addresses, Guilmette has spies lurking everywhere, watching for new spammers

"We have a global network of listening posts," says Guilmette "E-mail addresses that we put on different Web pages, that we have posting to newsgroups and mailing lists They are carefully disguised to look like normal e-mail addresses, but they actually forward every message that they receive back to our central database The spammers will never find them all."

Like most software engineers, Guilmette is confident that an acceptable solution is only a few lines of code away Spam technology, he says, is kid's stuff.

"As far as the arms race is considered -- their tricks vs our tricks -- I think their tricks have been extremely trivial," says Guilmette "It doesn't take a rocket scientist to figure out how to write a spam program But our defenses, or our tools or weapons, have, to date, been very primitive in response What I'm trying to do with Deadbolt is raise the bar to a much higher level. I agree with the people who assert that spammers may invent new tricks But right now they are only using a small handful of tricks, and they have no others Once we raise the bar on our side, the odds of them being able to jump over a higher bar is actually quite low "

Would that it were so Because without either the development of powerful defenses or punitive legislation, the spam onslaught is unlikely to slow The bulk e-mailers like to talk about responsibility -- some have even gathered together in consortiums aimed at coating their business with a respectable veneer But when pressed, most authors of bulk e-mail software disclaim any real responsibility for what is done with their programs In fact, Neil Albala, Forrest Dayton and Todd Farmer all immediately retreated to the identical clichés drawn from the libertarian arsenal of the National Rifle Association

"A person selling a gun is not responsible if someone goes out and shoots someone," says Albala "Almost any tool can be used irresponsibly Implying that I should be responsible is a bit of a stretch "

"A gun can kill a person," says Dayton "It is very simple to



go and kill someone But that's not what it was designed for "

Right And bulk e-mail software wasn't designed to bombard  
your mailbox with crap, either

Sept. 4, 1997



ALSO What you can do to stop spam

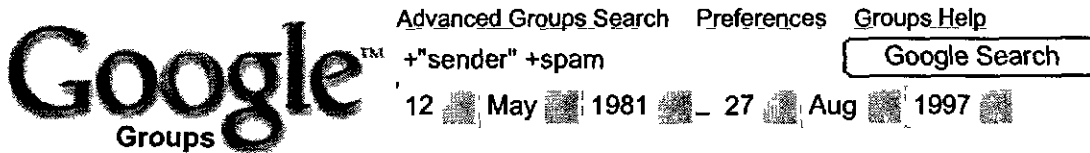
ILLUSTRATION BY ADAM McCAULEY

---

Is spam inevitable? How do you deal with it? Should the  
government step in -- or can technological tools solve a  
technological problem? Join the discussion in progress in  
Table Talk's Digital Culture area

s a l o n m a g a z i n e + a b o u t 2 1 s t + n e w s l e t t e r +





Groups search result 12 for + "sender" +spam

**Stop Spam. Free. Really.** • How? Amazon has a special \$20 promo for iHateSpam PC World's "Best Buy" • [www.sunbelt-software.com](http://www.sunbelt-software.com)

Sponsored  
Links

**Remove.Org - No Junk Mail** • Add Your Name To The Global Do Not Contact List And Stop Junk Mail • [www.Remove.Org](http://www.Remove.Org)

**Anti-Spam for Enterprises** • Spam filtering - No false positives 30 day trial - 1 hour to install • [www.frontbridge.com](http://www.frontbridge.com)

From [Bob Tiptrie \(tiptrie@cygnus.wa.com\)](mailto:Bob.Tiptrie@cygnus.wa.com)

Search Result 12

Subject: A way to stop spam messages

Newsgroups: [alt.privacy](#), [comp.bbs.misc](#), [alt.current-events.net-abuse](#), [spam.alt.spam](#), [alt.stop.spamming](#)

View [Complete Thread \(2 articles\)](#)

Date: 1997/04/26

[Original Format](#)

felbel@NOSPAMcsn.net (Fred Elbel) writes:

```
> On Fri, 25 Apr 1997 13:45:31 -0500, "Timothy J. Burns"
> <tburns@utelco.tds.net> wrote
>
>> I've recently begun posting to newsgroups, and suddenly my e-mail is
>> full of messages from spammers trying to get me onto every pyramid
>> scheme there is. What's the best way to stop this nonsense? I see some
>> people post with extra characters in their address, & some add the
>> language from a federal statute
>
> I believe the only way to stop it is to enact legislation against it. In
> the meantime, here are a lot of tips ...
```

There's another way, at least for Waffle users, as was recently posted.  
(copied below) It could be implemented for other systems also, I believe.

-----from comp.bbs.waffle:

```
] A proposal for a Waffle e-mail spam filter, provoked by a recent (ongoing)
] increase in the amount of e-mail spam
]
]
] I believe e-mail spam can be eliminated from Waffle systems in a relatively
] non-bothersome way by implementing the filter program described herein. I
] don't know the potential effects upon add-on mailreaders, however (this
] includes rebuilding the index!), and so I invite constructive comment.
]
] The scheme is based on the idea that e-mail will normally only be accepted
] from "registered" verified senders. The problem is that others cannot get
] through; this is solved by implementing a "call-back" reply which requires
] a simple, quick but manual operation on their part, to verify that the
] message is sent by a human, in order to get their messages delivered
]
] By their very nature spammers cannot afford the time to invoke the manual
] bypass for every one of their victims -- any spam which makes it through
] this filter is likely to have cost the spammer more time to send you the
] message than it takes for you to delete it
]
]
] PROGRAM DESCRIPTION
]
] The program as proposed is invoked after e-mail has been unpacked; it walks
] through the mailbox looking for e-mail which is neither from registered
] senders nor those on "probation" (verification of human entity in progress).
] The suspect e-mail is returned to sender with a note attached, then it is
] deleted from the mailbox.
]
```

```

] The sender is placed on "probation" for a predetermined amount of time,
] during which he can get one message through by adding a "key" sent to him
] in the note. This same message, and the "key", can be sent one time more
] in response to another piece of e-mail (i.e. sender gets a second chance)
] Thereafter, during the probationary period, unverified e-mail from that
] sender is summarily dropped.
]
]
] The procedure is outlined as follows:
]
] ---
] open probation file
] if file does not exist:
]     create empty probation file.
]
] // expire old entries
] for each entry in probation file:
]     if timestamp is earlier than current time:
]         delete entry
]
] open registry file
] if file does not exist:
]     // assume this is first time up and create registry
]     for each e-mail in mailbox:
]         if sender is not in registry.
]             add sender to registry.
]     exit
]
] // registry exists -- use it
] for each e-mail in mailbox:
]     {
]     if sender is not in registry file:
]         {
]         if sender is in probation file and timestamp is not FOREVER:
]
]             if e-mail first line matches key
]                 if mode is (N)otify:
]                     prepend 1st line to e-mail indicating sender's one-time approval.
]             else: // mode is (A)utomatic
]                 add sender to registry
]             delete entry in probation file.
]             else
]                 if retrycount is 0
]                     send same message (use message#) with key, timestamp and e-mail.
]                     set retrycount to 1.
]                     delete e-mail from mailbox.
]
]         else if timestamp is FOREVER:
]             delete e-mail from mailbox.
]
]         else // add him to the probation file
]             select and send message including key, timestamp, and e-mail
]             compute timestamp for expiration of entry
]             add sender, timestamp, key, message# and retrycount=0 to probation file.
]             delete e-mail from mailbox.
]         }
]     }
]
] close probation file.
] close registry file
] rebuild index
] exit
] ---
]
] A summary of the files involved (described below):
]

```

```

} nospam.NNN (NNN .= 000. 999)      Spam filter announcement files.
} nospam.key                          Spam filter keys file, ASCII text.
} nospam.reg                          Spam filter registry, ASCII text
} nospam.pro                          Spam filter probation registry
} nospam.exe                          The spam filter program.
}
}
} One difficulty with this system is that the user must manually add
} registered users to the registry file, though the first line of the e-mail
} warns him that he must do this. An alternative method is to add the sender
} automatically, at the risk of penetration by a spammer. That in turn could
} be countered by adding a FOREVER flag to the timestamp in the probation file,
} with all the implications thereto. A choice of operating mode could be made
} by a command-line switch.
}
} This program can either add verified senders to the registry directly, or
} it can notify the user when a sender is verified so the user can manually
} update the registry, depending upon his circumstances or level of paranoia.
} In either case a badly offending sending address can always be filtered
} out by adding him to the probation file permanently (timestamp FOREVER).
}
}
} FILTER ANNOUNCEMENT FILES.
}
} The note attached to the rejection is selected from one of a number of files
} created by the user, in different formats (to impede potential automated
} anti-spam-filter workarounds), all of which notify the sender of the action
} he must take to bypass the spam. Two examples follow (many more than two
} would be recommended) using different means to delimit the key:
}
} nospam.000
} Your message was rejected by an automatic spam filter which
} did not recognise your e-mail address. My apologies if this
} is an error. If you re-send your message before %d with
} the first line consisting of the text "%s" (without the quotes)
} your message will be delivered and I can then list you as an
} approved e-mail source.
}
} nospam.001
} This system has an automatic spam filter which rejected your
} message. You must re-send your message before %d and
} the first line must consist of the text:
} %s
} in order for your e-mail address to be registered as an approved
} e-mail source. Sorry for the inconvenience; spam has become a
} major nuisance here.
}
}
} The %d in these files is replaced by the timestamp (date & time) when
} the registry will expire.
}
} The %s in these messages is replaced by text selected at random from a
} filter "keys" file, also created by the user.
}
}
} FILTER KEY FILE.
}
} The keys could be numbers, words, phrases, or random letter combinations
} comprising a line of text. Having an assortment should help impede auto-
} detection of the key. When testing the returned key general whitespace-
} matching should be implemented since there are some editors with odd
} notions of space and tab usage. An example of a portion of a keys file:
}
} nospam key
} 1uchar
} clavis
} schlussel

```

```

] b-flat
] Francis Scott Key
] he came / out of England / with swift horse and heavy key
] 3 14159
] jnf(nou8344319@[/87hjdnpo~qpdbd9- poor sender who gets this'
]
]
] REGISTRY FILE:
]
] A registry file is simply a list of approved senders' addresses.
] dave@2001 com
] friend@xyzzzy edu
] neighbour'user
] Note: in a more "advanced" system the registry file, with additional
] fields, could be the address book, or vice versa. One could also
] integrate the probation file, with appropriate fields, but for the
] initial proposal I thought I'd keep this simple.
]
] THE PROBATION FILE.
]
] A probation file carries senders' addresses, the number of the message
] file used to reject their e-mail, the key, retry count, and the date
] and time of expiry:
] newcontact@there.com 001 "luchar" 0 1997.04.15 12:31:05
] badspammer@spam.net 002 "qwerty" 1 FOREVER
]
] PROGRAM SWITCHES.
]
] The executable would have several command-line switches:
] nospam exe -d<directory> -f<file> -t<hours> -m<mode>
] where:
] <directory> is the user's directory where the mailbox and
] "nospam" files are stored
] <file> is the user's mailbox filename.
] <hours> the number of hours for probation, = 1..65535.
] <mode> 'N'otify of approval, else automatically approve
]
] Thoughts?
-----
--
Robert Tiptrie
[Pithy quote here]
```

[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

©2003 Google





[Advanced Groups Search](#) [Preferences](#) [Groups Help](#)

challenge response "spam" group.comp security.

12 May 1981 - 27 Aug 1997

**Groups search result 2 for challenge response "spam" group.comp security.\***

[Sponsored Links](#)

**Anti-Spam for Enterprises** • Spam filtering - No false positives 30 day trial - 1 hour to install • [www frontbridge.com](#)  
**Eliminate Annoying Spam** • Want to kill time-wasting junkmail? Get iHateSpam PC World's "Best Buy" • [www sunbelt-software.com](#)  
**Spam Filter for Business** • World class spam elimination for businesses, enterprises & ISPs • [www mail-filters.com](#)

From [Douglas G Henke \(henke@ayesha.krall.org\)](#)

Search Result 2

Subject Re All Hail Emperor Lewis' (was Re status. shoppingplanet jan 20

Newsgroups comp.security.unix

[View Complete Thread \(19 articles\)](#)

Date 1997/02/20

[Original Format](#)

In article <199702200528.FAA02287@mauve.demon.co.uk> Ian Stirling  
 <root@mauve.demon.co.uk> writes:  
 > [ my article about munging email addresses in a manner which can be seen  
 > in the header and .sig of this article as well ]  
 >  
 > I immediately see a line with at, and dot in it. So I look up  
 > [ my real email address, which I'd truly appreciate it if you wouldn't  
 > post intact on Usenet' ] and bingo.

You've certainly got a point. In fact, a couple of them

- 1) it is fairly easy to figure out what is a sig automatically (especially if you have several different posts from the same author; I'm working on something to automatically trim .sigs from my incoming email)
- 2) it would be easy to write a program that would look for things like "user at host dot domain" in addition to "user@host domain"

Obviously, if \*everybody\* (or even a bunch of people) did things exactly the way I do, it would be worthwhile for spammers to code in a special case for it.

My argument is that if everyone makes up their own personal header munging / address presentation scheme, spammers should have to write so much ad-hoc code to recognize all the variants that it wouldn't be worth their while.

Practically anything can be parsed by a program once the format is known. The trick is coming up with something that 1) requires human intelligence to figure out (at least the first time), 2) is enough unlike what other folks do to foil existing address-reapers, and 3) is obvious enough to humans that it is still useful (unless you want to be anonymous).

I assert this works on the basis of the following I used to be henke@netcom.com. I posted to Usenet a lot, and got a heap of **spam**. In November of 1996, I moved from netcom to phoenix and at the same time started munging headers on all my Usenet posts. The only email **spam** I have gotten at phoenix so far has been what has been forwarded from my netcom account. As far as I know, phoenix has absolutely no email-**spam** filters of any sort in place

I also contend that header mangling is only useful as part of a larger campaign of **spam** avoidance. I mangle headers. I blacklist certain domains with procmail. I run "trap" address designed to garner **spam**, and use what they receive as fuel for the blacklist I am experimenting with a CRAB-type system (first-time email correspondents are given a **challenge**, and must send a **response** in order to have their mail get through).

IMHO, the real solution has to be at the source: providers blocking \*outgoing\* **spam**, denying known spammers net access, or doing enough damage (social, monetary or physical) to spammers in the real world to eliminate their



ability to **spam**, thereby providing a deterrent for others.

> If I'm sufficiently confident, I may not even bother looking it up

Makes sense, generate everything you even \*think\* might be an address, try them all and remember which ones work. Wrong guesses carry no penalty except for a miniscule amount of time and bandwidth (That brings up another possible flaw in my scheme. my login on ayesha is the same as my mailbox name; the phoenix domain name appears in NNTP-Posting-Host , Path and maybe other places.)

> Lots of people do it, by replacing . with dot, and similar mungings.  
> This is a thing computers are good at.

I literally do hack Perl for food, so I know this I also know that people are way better than programs at figuring out new stuff the first time around. For example:

My backwards address net dot phoenix at henke  
My reversed address: ten.xineohp.ekneh

```
M m   r   n p n n
y a a e   k h i e
  i d s h e o x t
    e l d s e @ e .
```

My login name is my last name. My mailhost is named for a mythic bird of fire, or a city in Arizona. Top-level domain is "net."

The big problem I see with the whole thing is that, in many non-technical newsgroups, if "reply" doesn't work immediately, newbies will just hit "followup" and post drivel that everybody has to read instead of just me.

--

Headers munged to avoid email **spam**. There are plenty of clues in the article for you to find my real address for email replies'

[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

©2003 Google





[Advanced Groups Search](#) [Preferences](#) [Groups Help](#)

Google Search

Viewing message <56luge\$6on@bertrand.ccs.carleton.ca>

**Stop Unsolicited Email** • Try Earthlink for Award Winning Pop-Up & Spam Blocker Software • [www.earthlink.com](http://www.earthlink.com)

Sponsored

**Stop Spam Free. Really** • How? Amazon has a special \$20 promo for iHateSpam PC World's "Best Buy" • [www.sunbelt-software.com](http://www.sunbelt-software.com)

Links

**DynaComm Lmail** • Spam filtering from FutureSoft Download your trial today • [www.dcseries.com](http://www.dcseries.com)

From David F. Skoll (dfs@doe.carleton.ca)

Subject: How to make SURE a human is sending you mail (was Re: Random e-mails)

Newsgroups: [news.admin.net-abuse.usenet](#)

[View Complete Thread \(7 articles\)](#)

Date: 1996/11/17

[Original Format](#)

> On 15 Nov 1996 00:41:49 GMT, Brandon Hume <hume@isisnet.com> wrote  
> >

> >How is a program supposed to distinguish junk mail from mail that you  
> >actually want? What would your ISP's customers DO if they even HEARD  
> >that they were filtering your email? Nothing good, I assure you

I've thought long and hard about this and have come up with a simple solution which will stop spamming. All I need is the time to implement it -- if someone else wants to do it, go ahead!

Here's the scoop. I want to make SURE a person is sending me e-mail and not a computer. I distinguish three kinds of e-mail addresses

1 - Trusted addresses -- these are addresses of friends or associates  
Mail from these addresses gets sent to my mailbox

2 - Known bad addresses -- these are addresses which I know are used by spammers. Mail from those addresses gets silently discarded

3 - Unknown addresses. These are the tricky ones. When mail is received from an unknown address, my autoresponder replies with a "puzzle." This puzzle is a random challenge that is very easy for a human to respond to, but next to impossible for a computer. For example, the reply could look like this

I apologize for the inconvenience, but due to the prevalence of spamming, I ask that you authenticate yourself as a human. Please re-mail your message adding the following header.

X-Not-Spam: xxxxx

replacing the "xxxxx" by the English name of the fourth month, all in lower case

Once again, I apologize for the inconvenience and trust that you are understanding. Regards,

Of course, you need a file of many simple "puzzles" which are inserted in the reply appropriately. The program keeps track of which puzzles have been sent to whom, and only allows you to see mail from people who successfully authenticate themselves. This is trivial for a person, but a nightmare for someone who wants to mass-mail thousands of people, each of whom has a different puzzle. After a couple of authentication failures, the program would mark the address as "bad" and discard mail from it, breaking potential bounce cycles.

Mail from "postmaster" is problematic -- you really might want to see messages about mail that you sent that has problems, but you don't want

to simply allow all "postmaster" messages through -- spammers will simply fake that as their source address. So, you need to add a random string to all \*outgoing\* messages. If outgoing mail bounces, the mail system will include the message body (or at least the header). The program can verify that mail purportedly from postmaster contains the string and isn't unsolicited.

Of course, if you send mail to someone, you should include instructions for authentication to reduce the inconvenience for that person if he/she wishes to reply.

The advantage of this scheme over others is that most other junk filters allow the first message through and force you to manually remove spam addresses. This scheme treats all unknown addresses with suspicion, which makes it slightly inconvenient for legitimate users, but next to impossible for spammers to get through. It can be refined ad nauseum by adding timers to passwords, checking for addresses attempting mail bombing, etc.

Regards,

David

P S I've deliberately set a bad reply-to address, I don't want to be mail-bombed for this! .-)

[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

©2003 Google



# Welcome to...

## MailCircuit.com

[Email Hosting Services](#)   [Gateway Servers](#)  
[Reseller Plans](#)   [Spam and Virus Free Solutions for Exchange S](#)  
[Complete Mail Systems](#)   [Corporate Plans](#)



MailCircuit User ID

Password

|                              |
|------------------------------|
| ■ MailCircuit User Accounts  |
| <b>Signup</b>                |
| ■ MailCircuit Included Tools |
| <b>Spam Filtering</b>        |
| <b>Virus Scanning</b>        |
| <b>Mail Routing</b>          |
| <b>Account Features</b>      |
| ■ Customer Service           |
| <b>FAQ's</b>                 |
| <b>Contact Us</b>            |
| <b>User Feedback</b>         |
| <b>Email Client Setup</b>    |
| ■ Other Services             |
| <b>White Pages</b>           |
| <b>Yellow Pages</b>          |
| <b>Internet Access</b>       |
| <b>Server Hosting</b>        |
| ■ Legal                      |
| <b>Terms of Service</b>      |
| <b>Your Privacy</b>          |
| <b>Refund Policy</b>         |
| ■ Glossary                   |
| <b>Email Definitions</b>     |
| <b>Internet Definitions</b>  |

### MailCircuit's Email HandShake Verification™ and Spam Filter Process

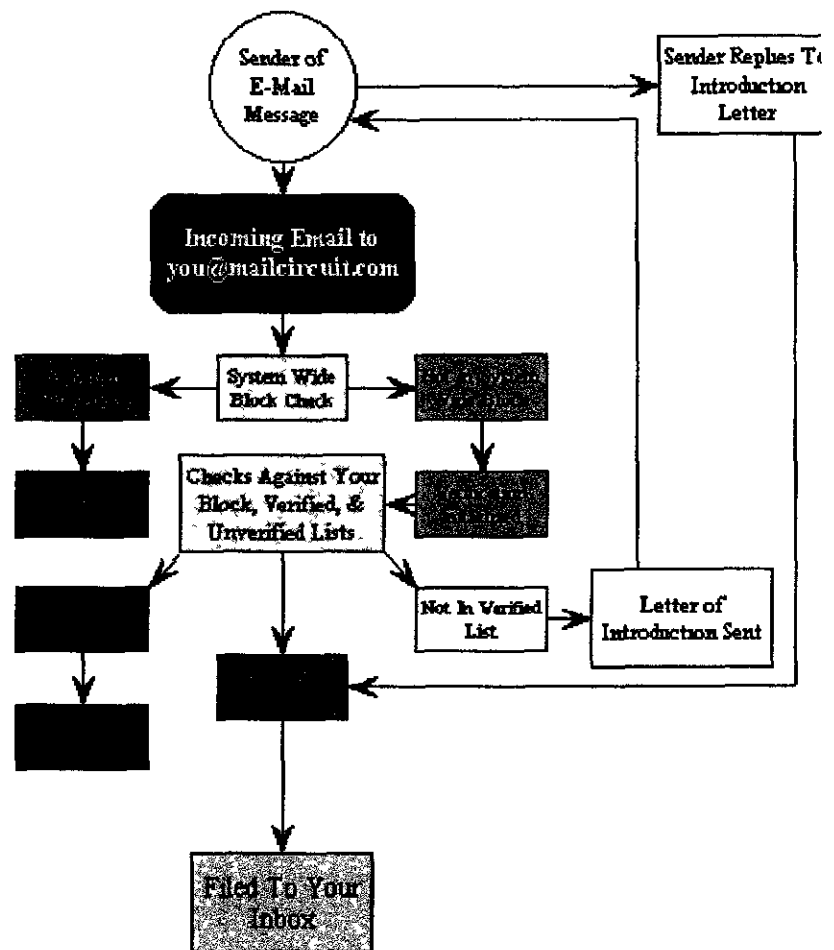
Our most exciting feature!

Our Patent Pending Email Mail Verification Program™ stops 100% of unapproved mail.

Here's how it works:

Of course, we first apply common spam filtering and blocking techniques based on rules-based known-spam criteria. Should you elect to activate our HandShake Verification option, to achieve 100% protection, then all remaining unapproved incoming message traffic is filtered as follows: A check is made to see if the message is from a trusted sender previously approved by you. If yes, the message is delivered to your inbox. However, if the message is from an unverified address, our system places the message in a special user-account hold-queue and sends an encoded verification message to the sender. This message asks them to reply to a uniquely-encoded verification code to our system. If they make a positive reply (handshake), the original message is retrieved from the hold-queue and then delivered into your inbox. The verified incoming address is added to a trusted list of senders (which you can add or block at any time thereafter via our easy web interface). However, if the sender does not reply to the encoded verification message within a set time (chosen by you in your User Settings) then the message is deleted and the sender's address is placed on your unverified or "block" list. Of course, automated messages from email lists and other non-human senders, at any time go into the hold-queue and you can promote a sender manually to verified using our convenient web interface to your email account.

Here is the process flow in detail:



Click on the picture for a detailed description.

Copyright © 1996- 2003 MailCircuit.com





We pursue all legal avenues to get your valuable sales message heard!

EZSPAM (tm) only \$299.95 (standard) or \$399.95 (gold).

See <http://www.ez137spam.com/> for further information

Get your EZSPAM (tm) (patent pending) today!

-----  
For the humor impaired: this is a \*satire\*. It is \*not\* real (yet).

Unlikely though? I think not. :-( Exaggerated? Not by much. Some of these things ('personalised messages', 'layout permute', 'teasers' and 'ISP diversity') are already being used by spammers. Computer viruses already have full permutation engines. Also, look at the level of sophistication that conventional advertising is operating at.

Not much in this post is difficult to do, particularly once organized crime gets involved. Some people may be unhappy that I'm advertising these options to spammers. Sorry, but they know about most of them already. Security through obscurity is rarely a good idea long term.

The only reason it hasn't happened yet is that only a vanishingly small percentage of net sites are/were filtering until AOL started recently. It just wasn't worth a spammer's trouble. Just give it time. All it takes in one person in the whole world to write EZSPAM and all of the thousands of conventional filtering setups that netizens have developed will be for naught.

Conventional filtering is just going to become less and less effective, make more and more mistakes leading to lost email and poor communication and eventually sink into the noise. Only full human level AI, a secretary if you like, can work out where any particular email message is on the spectrum between junk and valuable. Even then mistakes are made. Filtering works, but in the long term only when the sender is cooperating.

The advantage of the **challenge-response** anti-spam 'bot (CRAB ;-)) is that it places the onus on the sender to have human level AI, not the recipient and email is not lost or misplaced. CRAB's have problems but they'll work medium term and the overhead (first time contact slightly messier) is minimal. Sure I'd like to go back to when the net was young, netizens could be trusted and anybody could email anybody with no worries about foolish filters losing email, delaying email or letting unsolicited junk through and idiotic CRAB's wasting their time. Those days are gone.

Julian Byrne <<http://kryten.eng.monash.edu.au/gspam.html>>

[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

©2003 Google





[Advanced Groups Search](#) [Preferences](#) [Groups Help](#)

Google Search

Groups search result 1 for

**Eliminate Annoying Spam** • Want to kill time-wasting junkmail? Get iHateSpam PC World's "Best Buy" • [www.sunbelt-software.com](http://www.sunbelt-software.com)

Sponsored  
Links

**Spam Filter for Business** • World class spam elimination for businesses enterprises & ISPs • [www.mail-filters.com](http://www.mail-filters.com)

**Smart Inbox Assistant** • Learns Moves Spam, Newsletters Free Trial Sorts Outlook Email • [www.openfieldsoftware.com](http://www.openfieldsoftware.com)

From Julian Byrne (no\_junk\_email@any\_where)

Subject My spamblock, Was Thwarting UCE address culling programs

Newsgroups [news.admin.net-abuse.email](#), [comp.mail.sendmail](#),

[comp.security.unix](#)

Date 1997/01/19

[View Complete Thread \(11 articles\)](#)

[Original Format](#)

David Richards wrote

> [Description of spam block scheme]

I've been using a similar email spam block 'bot experimentally for months.

It must be good; I've received hate mail from spammers concerning it -)

How it works

- Addresses on a 'blocked' list are bounced with a rude message
- Addresses on a 'go' list are delivered as usual
- Any 'unknown' address is bounced with a note saying that the sender needs to put a code in the 'Subject' heading and their email will get through
- If a message is received with a valid code the corresponding address is put on the 'go' list
- Any address I send to is automatically placed on the 'go' list so that anybody who replies to me never sees the 'bot
- Bounces and double bounces must be handled very carefully

Upside

- \*Zero\* spam
- Much reduced thoughtless throwaway email as people get a second chance to decide not to send me an email message.
- No maintenance, invisible to me and my regular correspondents
- Doesn't matter what fake domains, IP addresses and relays spammers use.
- Email addresses I correspond with are verified
- Once installed easy for a naive user to understand and use
- Several of my correspondents have asked for a copy
- Doesn't require the whole net cooperating to work

Downside

- People contacting me for the first time have to put a code in the subject heading
- People who use multiple email addresses get aggravated I suspect Rahul would hate it :-)
- A pain for postmasters, support people, mailing list maintainers etc who deal with large volumes of email to one-off correspondents
- Can anybody think of a long term solution to this?
- Complicated to install - inbound, outbound, bounce and double bounce messages must all be handled
- Doesn't eliminate the machine overhead of the spam email

I believe a development of this could be one medium term solution to the email spam problem. Currently the 'bot asks for a code that a spammer could automatically deal with but by replacing the code with arbitrary question[s] for each user (eg 'What color do red and yellow paints mixed together give? Put it in the subject heading') then I can't see any way that a spammer could automatically handle it. It depends on the fact that a human being will always be able to ask and answer a question that an AI can't, for the near future anyway.

If you were a spammer could you beat it?

I'm not releasing the 'bot code at this stage as I still regard it as mostly experimental. It's written mainly in C/gdbm and runs under unix with gmail.

Incidentally, Dan is right about fully automatic filters: they are only a stopgap. That's why I stress AUP's and other non-filter based solutions on my web page. When enough people start filtering the spammers will simply change their forgeries more often. If need be they can even change it on the fly during a single spam run.

Julian Byrne <<http://kryten.eng.monash.edu.au/gspam.html>>

[Google Home](#) - [Advertise with Us](#) - [Business Solutions](#) - [Services & Tools](#) - [Jobs, Press, & Help](#)

©2003 Google